

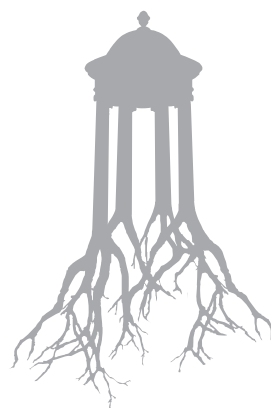
UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



## Simulación visual de la iluminación → Teoría. Técnicas. Análisis de casos

Javier Monedero





UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



iniciativa  
digital politècnica  
Publicacions Acadèmiques de la UPC

→ **UPCGRAU**

# Simulación Visual de la Iluminación → Teoría. Técnicas. Análisis de casos

Javier Monedero

Primera edición: diciembre de 2015

© Javier Monedero, 2015

© Iniciativa Digital Politècnica, 2015  
Oficina de Publicacions Acadèmiques Digitals de la UPC  
Jordi Girona 31,  
Edifici Torre Girona, Planta 1, 08034 Barcelona  
Tel.: 934 015 885  
[www.upc.edu/idp](http://www.upc.edu/idp)  
E-mail: [info.idp@upc.edu](mailto:info.idp@upc.edu)

DL: B. 21036-2015  
ISBN: 978-84-9880-545-1

Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sólo puede realizarse con la autorización de sus titulares, salvo excepción prevista en la ley.





# Índice

<b>TEORÍA</b>	<b>19</b>
<b>1. Escenarios reales</b>	<b>21</b>
1.1 La luz, los objetos y el ojo	21
1.2. La luz. Nociones generales. Unidades radiométricas y fotométricas	21
Cantidades radiométricas y fotométricas	22
Las cuatro unidades fotométricas principales	22
Luminancia, radiancia, irradiancia, reflectancia, brillo	27
1.3 Tipos de luces	28
El Sol. Radiación solar	29
El cielo. Radiación celeste	31
El fenómeno de la dispersión	36
La perspectiva aérea	38
La luz nocturna	40
Luces artificiales	42
Lámparas y luminarias	43
Color y calidad lumínica	43
1.4 Luces y sombras. Calidad y función visual de las sombras	45
1.5 Los objetos. Modulación de la luz por los objetos	48
1.6 El ojo y el sistema visual humano	49
La estructura del ojo	49
Mecanismos neuronales de inhibición lateral	52
Receptores cromáticos profundos. El córtex visual	55
Una visión global del cerebro	57
1.7 La formación de imágenes	58
Límites geométricos. Capacidad inicial de resolución del sistema visual	59
Poder de resolución y difracción. Círculos de confusión o círculos de Airy	62
Funciones de dispersión en el ojo	64
Detección y reconocimiento. Agudeza Vernier	64
1.8 La formación de imágenes. Límites de discriminación de valores	66
Condiciones de adaptación	66
La discriminación de luminancias. Leyes de Weber/Fechner	67
La cuantificación del contraste	69
La integración de la resolución y los valores. Funciones de transferencia	71
1.9 Rangos dinámicos. Escalas de valores	74
Umrales absolutos y relativos	74
Rangos dinámicos	79
La construcción de escalas de valores	81
Las bandas de Mach	85
El sistema de zonas	86



<b>2. Escenarios virtuales.....</b>	<b>89</b>
2.1 La luz, los objetos y el ojo en un entorno virtual .....	89
2.2 Luz virtual. Unidades. Calidad lumínica .....	89
Unidades .....	90
La calidad lumínica.....	91
2.3 Tipos de luces .....	92
Luces virtuales simples .....	92
Luces básicas con propiedades de área y volumen (extensas).....	96
Luces fotométricas .....	96
Luz solar. Luz celeste. Sistemas de luz diurna .....	98
Sistemas de luz diurna (integración de luz solar y luz celeste).....	100
Simulación de luminarias .....	100
2.4 Luces y sombras. Introducción a los tipos de sombras y sistemas de cálculo .....	101
Volumen de sombra ( <i>shadow volume</i> ) .....	101
Sombras por mapa ( <i>shadow map</i> ) .....	102
Sombras por trazado de rayos ( <i>ray traced shadows</i> ) .....	103
Sombras de área.....	104
Sombras suavizadas por luces secundarias y por cálculos de iluminación global ..	104
Sombras de objetos transparentes .....	105
Otros tipos de sombras. Luces sin sombras. Luces negativas .....	105
2.5 Objetos virtuales. Los objetos virtuales como moduladores de luz.....	106
2.6 Ojos/cámaras virtuales.....	107
2.7 La formación de imágenes virtuales. Límites geométricos .....	109
Límites geométricos. Capacidad de resolución de dispositivos virtuales.....	109
2.8 La formación de imágenes. Rangos dinámicos y escalas de valores.....	112
Rangos dinámicos de dispositivos artificiales .....	113
Rangos de dispositivos de entrada. Cámaras digitales. Escáneres .....	113
Rangos de dispositivos de salida. Monitores. Impresoras .....	120
Construcción de escalas en dispositivos analógicos. La densitometría.....	123
Construcción de escalas en dispositivos virtuales. El factor <i>gamma</i> .....	126
Formatos de alto rango dinámico (HDR). Introducción. Conceptos básicos y aplicaciones.....	132
Distribución de valores en formatos HDR. Tone mapping.....	137
<b>3. Sistemas de cálculo de iluminación.....</b>	<b>141</b>
3.1 Introducción.....	141
Procedimientos básicos y marco general del proceso de cálculo.....	141
Sistemas de iluminación local .....	142
3.2 Sistemas de iluminación global .....	142
Resumen de la evolución histórica.....	142
Ventajas y desventajas de los métodos disponibles .....	143
3.3 <i>Ray tracing</i> clásico (1980).....	145
Métodos precursores. Cálculos de visibilidad y sombras (1968, 1971) .....	145
Métodos clásicos de ray tracing (1980).....	146
Ray tracing distribuido (1984) .....	146
3.4 Radiosidad .....	147

	Método general (1984).....	147
	Técnicas de optimización.....	149
	Recolección final (FG) en radiosidad .....	150
3.5	La <i>rendering equation</i> .....	150
	La <i>rendering equation</i> (1986).....	150
	Coordenadas hemisféricas. Radiancia. Irradiancia.....	153
	La función de reflectancia (BRDF) .....	154
	<i>Path tracing</i> (1986).....	156
	Estrategias de cálculo. <i>Forward tracing</i> . <i>Backward tracing</i> .....	157
	Función de respuesta ( <i>response</i> , <i>importance function</i> ). <i>Measurement equation</i> ...	160
3.6	Métodos de Montecarlo. Métodos generales de <i>ray tracing</i> .....	161
	Orígenes del método de Montecarlo .....	161
	Métodos básicos de Montecarlo en simulaciones de iluminación.....	164
	Métodos de optimización .....	165
	<i>Quasi Monte Carlo</i> .....	166
	Cálculo de la iluminación directa e indirecta .....	167
	<i>Ray tracing</i> generalizado. Métodos de optimización. Estructuras intermediarias ...	169
3.7	Métodos de pases múltiples ( <i>multipass</i> ). Métodos <i>biased</i> .....	171
	Métodos combinados de <i>ray tracing</i> y radiosidad (1986).....	172
	<i>Bidirectional path tracing</i> (1993).....	172
	<i>Irradiance cache</i> (1988, 1992) .....	173
	Proyección de fotones ( <i>photon mapping</i> , 1995).....	176
	Proyección de fotones y cáusticas .....	178
	<i>Final gather</i> .....	179
	<i>Metropolis light transport</i> (1997).....	182
3.8	<i>Image based lighting</i> (IBL) .....	184
	Fundamentos .....	184
	Procedimiento general .....	186
3.9	<i>Ambient occlusion</i> .....	188
3.10	Otros métodos.....	189
	Importons .....	189
	<i>Irradiance particles</i> .....	191
	<i>Brick maps</i> . <i>Point clouds</i> . <i>Point based GI</i> .....	191
	<b>TÉCNICAS. ANÁLISIS DE CASOS</b> .....	195
	<b>4. Recursos básicos y complementarios</b> .....	197
4.1	Recursos básicos .....	197
	Las tuberías de la representación .....	197
	Tarjetas gráficas. La GPU.....	197
	Direct3D y OpenGL. <i>Shaders</i> . Otras tecnologías recientes .....	199
	Las tuberías de la representación .....	202
	Los cálculos de iluminación y la <i>pipeline</i> .....	204
	Cálculos estandarizados .....	204
	Iluminación global con la GPU .....	206
	Aspectos generales. Ventajas y limitaciones del uso de la GPU. <i>Rasterization</i> frente a <i>Ray tracing</i> .....	206



	Múltiples luces (VPL). <i>Instant radiosity</i> . <i>Lightcuts</i> . Otras variantes .....	209
	<i>Precomputed Radiance Transfer</i> (PRT). Otros métodos de precómputo .....	213
	<i>Ray tracing</i> con la GPU .....	215
	<i>Photon mapping</i> .....	218
	<i>Ambient occlusion</i> .....	220
	<i>Final gather</i> e <i>irradiance caching</i> .....	222
	<i>Aliasing</i> y <i>antialiasing</i> .....	226
	Conceptos fundamentales .....	226
	Supermuestreo y submuestreo. Muestreo adaptativo .....	228
	Ejemplos de configuraciones básicas .....	229
	<i>Unified sampling</i> . <i>Progressive rendering</i> .....	230
	Filtros. Filtros usados en cálculos de iluminación .....	232
	Organización de la escena .....	233
	Aspectos generales .....	233
	Material de superposición ( <i>override material</i> ) .....	233
	Cámaras .....	234
4.2	Uso de imágenes HDR en cálculos de iluminación .....	236
	Imágenes HDR .....	236
	Métodos generales de obtención de imágenes HDR .....	236
	Grabación de series de fotografías con varias exposiciones .....	237
	Montaje y generación de archivos HDR a partir de series de fotografías .....	240
	Imágenes panorámicas .....	241
	Mapas de entorno panorámicos .....	242
	Tipos principales .....	243
	Software de generación de panoramas .....	246
	Transformaciones de panoramas .....	247
	<i>Skydomes</i> y <i>skyboxes</i> .....	249
	Métodos de grabación y montaje de panoramas HDR a partir de escenarios reales .....	251
	Grabación de series de fotografías solapadas .....	251
	Grabación por medio de una bola especular ( <i>mirror ball</i> ) .....	255
	Grabación con cámaras especiales .....	259
	Montaje y generación de HDR panorámicos .....	260
	Métodos de grabación y montaje a partir de escenarios virtuales .....	262
	Procedimiento general con <i>Terragen</i> .....	262
	Montaje externo .....	266
	Procedimiento general con <i>3ds Max</i> o programas similares .....	267
	Montaje externo .....	270
4.3	Edición y ajuste de imágenes HDR .....	270
	Edición .....	270
	Análisis previo .....	270
	Herramientas y limitaciones de edición .....	271
	Ajustes .....	272
	Ajustes básicos .....	272
	Ajustes locales .....	273
	Ejemplos .....	274
	Ejemplo 1 .....	274
	Ejemplo 2 .....	275

4.4	Reproyección de valores ( <i>tone mapping</i> ). Ajustes de <i>gamma</i> . Control de exposición .....	276
	Visualización LDR de archivos HDR .....	276
	Reproyección de valores ( <i>tone mapping</i> ) .....	276
	Métodos automáticos .....	277
	Ajustes de <i>gamma</i> .....	279
	Tipos generales de ajustes .....	279
	Ajustes locales de <i>gamma</i> .....	280
	Control de exposición .....	282
	Problemas con fondos y control de exposición .....	283
	Fondos demasiado oscuros o claros .....	284
<b>5.</b>	<b>Iluminación directa. Luces y Sombras .....</b>	<b>287</b>
	Nota sobre los ejemplos .....	287
5.1	Luces simples puntuales, focales, directas .....	288
	Interfaz y métodos de creación y modificación .....	288
	Parámetros básicos de luces simples .....	289
	Atenuación con luces simples .....	291
	Recursos adicionales. Exclusión. Afectar a difuso. Afectar a especular .....	293
	Luces simples extensas .....	293
	Luces extensas planas .....	294
	Luces extensas de volumen .....	294
5.2	Luces fotométricas .....	294
	Parámetros básicos de luces fotométricas .....	294
	Unidades de luces fotométricas .....	295
	Unidades y control de exposición .....	297
	Temperatura de color. Plantillas .....	297
	Tipos de distribución. Archivos IES. Otros archivos .....	297
5.3	Sombras .....	301
	Parámetros generales .....	301
	Tipos de sombras .....	302
	Sombras por mapa ( <i>shadow maps</i> ) .....	302
	Sombras por <i>ray trace</i> .....	305
	Sombras de área .....	306
	Sombras de volumen .....	307
5.4	Sistemas de luz diurna .....	308
	Procedimientos generales. Creación y especificación de datos climatológicos .....	308
	Luz solar. Tipos y propiedades .....	311
	Luz celeste. Tipos y propiedades .....	312
	Luz celeste. Modelos de cielos .....	314
	Perspectiva aérea .....	317
	Mapas de entorno .....	319
	<i>Sky portals</i> .....	320
	Control de la temperatura de color en <i>sky portals</i> .....	323
5.5	Luminarias y objetos luminosos .....	323
	Luminarias .....	323
	Tres ejemplos .....	323

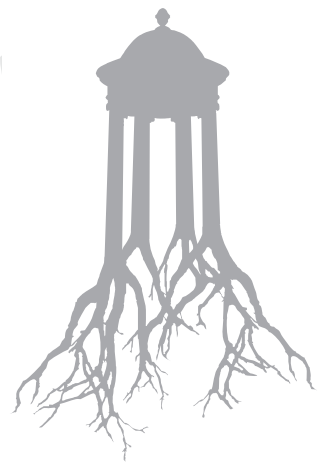


Objetos irradiantes .....	326
Efectos de volumen de luz y de resplandor ( <i>glow, glare</i> ).....	328
<b>6. Sistemas y configuraciones de cálculo de iluminación global .....</b>	<b>333</b>
6.1 Iluminación directa e iluminación indirecta.....	333
Sistemas de iluminación local y de iluminación global.....	333
Tipos de organización de la escena y tipos de luces en ambos sistemas .....	333
Tipos de cálculo de sombras en ambos sistemas.....	334
Dependencia del punto de vista y de la resolución. Métodos de filtrado y <i>antialiasing</i> .....	334
Ventajas de los sistemas de iluminación local .....	335
Ventajas de los sistemas de iluminación global y sumario de los descritos en este capítulo.....	335
Principales aplicaciones (motores de <i>render</i> ) disponibles en 2014 .....	337
RenderMan (1988, Pixar).....	339
Mental Ray (1989, mental images, Nvidia).....	340
Cinema 4D (1993, Maxon) .....	341
Lightwave 3D (1994, Newtek) .....	342
Mantra con Houdini (1996, Side Effects Software).....	342
V-Ray (1997, Chaos Group).....	343
3Delight (2000, DNA Research) .....	344
FinalRender (~2001, Cebas).....	344
Modo (2004, The Foundry).....	345
Maxwell (2006, Next Limit) .....	346
Arnold (2009, Solid Angle).....	347
Octane (2011, Ottoy) .....	348
Clarisse (2012, iFX Isotropic) .....	349
Lagoa MultiOptics (2013, Lagoa) .....	349
Escenario utilizado en los análisis de este capítulo .....	350
6.2 Radiosidad .....	351
Radiosidad con 3ds Max.....	351
Preparación. Escenario utilizado. Sumario del método.....	352
Primera fase: calidad inicial.....	352
Segunda fase: mallado.....	353
Tercera fase: <i>regather</i> .....	354
Control de las características y densidad de la malla .....	355
Control de la fuga de color con radiosidad .....	356
Radiosidad con otros programas .....	357
6.3 <i>Path tracing</i> y otros métodos básicos de Montecarlo.....	358
Métodos de <i>path tracing</i> con <i>iray</i> de <i>mental ray</i> .....	358
Descripción general.....	358
Parámetros .....	360
Ejemplo 1 (interior) .....	360
Ejemplo 2 (exterior) .....	362
<i>Path tracing</i> con V-Ray ( <i>light cache</i> ) .....	363
Procedimientos de configuración de iluminación global en V-Ray .....	363
Descripción general.....	364

Parámetros generales .....	365
Análisis .....	367
“Fuerza bruta” o DMC ( <i>Deterministic Monte Carlo</i> ) con V-Ray .....	368
Descripción general.....	368
Parámetros. Análisis.....	370
6.4 Irradiancia y Final Gather.....	371
Irradiancia con V-Ray .....	371
Descripción general.....	371
Parámetros generales .....	372
Análisis .....	373
Final Gather con mental ray .....	375
Procedimientos generales de configuración de iluminación global con	
<i>mental ray</i> .....	375
Descripción general.....	376
Parámetros .....	378
6.5 Fotones. Combinación de irradiancia y fotones. Cáusticas .....	382
Fotones con mental ray .....	383
Descripción general.....	383
Parámetros generales .....	383
Análisis .....	383
Combinación de irradiancia y fotones con mental ray.....	386
Fotones con <i>mr sun</i> .....	387
Control de la fuga de color ( <i>color bleeding</i> ) con fotones .....	387
Fotones con V-Ray.....	387
Parámetros generales .....	387
Análisis .....	390
Combinación de irradiancia y fotones con V-Ray.....	390
Cáusticas.....	391
Cáusticas con mental ray .....	392
Descripción general.....	392
Parámetros generales .....	392
Análisis .....	394
Ejemplos.....	395
Problemas característicos .....	397
6.6 Métodos de optimización.....	398
Reutilización de mapas de irradiancia, final gather y fotones .....	398
Uso de <i>importons</i> e <i>irradiance particles</i> .....	400
6.7 IBL ( <i>Image Based Lighting</i> ).....	402
Procedimientos comunes a las diferentes aplicaciones.....	402
Tipos de aplicaciones y tipos de mapas HDR. Conversión entre formatos.....	402
Combinación de mapas HDR y LDR para procesar fondos, entornos y reflejos..	402
Entornos virtuales o integrados como objetos hemisféricos .....	403
Problemas característicos .....	404
IBL con mental ray .....	405
Carga y configuración de un archivo HDR o EXR.....	405
Alternativas de configuración para IBL.....	407
Escenario y mapas utilizados en los cuatro primeros ejemplos que siguen .....	410
Ejemplo 1. Un mapa. Con FG y un objeto envolvente hemisférico irradiante .....	411

Ejemplo 2. Un mapa. Con FG y <i>Skylight</i> .....	414
Ejemplo 3. Un mapa. Con <i>iray</i> .....	415
Ejemplo 4. Un mapa. Con IBL nativo .....	416
Mapas utilizados en el último ejemplo que sigue .....	417
Ejemplo 5. Dos o Tres mapas. Iluminación, fondo, reflejos.....	419
Smart IBL (sIBL) .....	421
Uso de IBL con otras aplicaciones .....	421
6.8 Oclusión ambiental ( <i>Ambient Occlusion</i> ) .....	422
<i>Ambient Occlusion</i> como sistema de cálculo de iluminación sin luces .....	422
Descripción general.....	422
Parámetros .....	422
Ejemplo.....	424
6.9 Análisis y medición de la iluminación.....	426
Análisis de la iluminación con radiosidad .....	427
Recursos .....	427
Análisis de la iluminación con <i>mental ray</i> .....	427
Recursos .....	427
Precisión de los análisis .....	428
Parámetros principales.....	429
Ejemplo 1. Galería. Iluminación diurna.....	430
Ejemplo 2. Galería. Iluminación con luces artificiales .....	433
<b>Referencias</b> .....	435
<b>Índice analítico</b> .....	443









## Introducción

La primera parte de esta introducción es común a dos libros que se publican en paralelo: *Simulación visual de la iluminación* y *Simulación visual de materiales*. A partir de aquí, los desarrollos se bifurcan aunque, en más de un ocasión me veré obligado a remitir el uno al otro y el otro al uno. Y, en algún caso, será inevitable alguna repetición, pues las fronteras se solapan.

La creación de escenarios virtuales abarca muchos temas. Pero los dos principales son la representación o simulación de los materiales y de la iluminación. La interconexión es obvia pues, si no hay materiales que redistribuyan la iluminación, no podemos analizarla y, si no hay iluminación, poco podremos decir de las propiedades visuales de los materiales. Pero los problemas teóricos y técnicos implicados divergen.

Mi intención, que quizás atraiga a menos lectores aunque, espero, más exigentes, no ha sido elaborar un manual sobre cómo utilizar ciertas técnicas, sino situar estas técnicas en un contexto más amplio que pase, en primer lugar, por analizar los escenarios reales antes de intentar replicarlos por medios virtuales. Y esto supone un análisis extenso de las propiedades generales, físicas y perceptivas que se dan en la realidad. Y, en segundo lugar, por lo que respecta a los escenarios virtuales, explicar los conceptos teóricos que hay detrás de las técnicas. Pues creo que el único modo de aplicar las técnicas de un modo adecuado es comprender bien la base teórica en que se apoyan.

Aunque parezca obvio que sea imprescindible analizar a fondo las peculiaridades de las cosas reales antes de intentar simularlas por cosas virtuales, a lo largo de muchos años de docencia he constatado que esto no es así. Y, hasta cierto punto, es lógico que no lo sea, pues este análisis es complejo y abarca

dos maneras muy distintas de abordarlo: una científica y otra artística. La obra de Leonardo da Vinci es un paradigma de lo que se puede conseguir cuando se atiende a estos dos extremos. En una época en que parece que todo consiste en saber cuál es el botón que hay que apretar me parece saludable intentar, por difícil que sea, recuperar esta tradición.

Estas son las razones de esta división en técnicas de simulación de iluminación, por un lado, y de simulación de materiales por otro. Y ¿por qué “simulación” en lugar de, por ejemplo, “representación”? No hay razones muy claras para elegir uno u otro término y confieso que he dudado entre ambos. Pero el hecho es que el término *simulación* se ha utilizado de modo preferente en los últimos años, por un lado para diferenciarse de los métodos tradicionales y, sobre todo, para recalcar el hecho de que los resultados se basan en procesos internos que implican bastantes más cosas que la simple proyección de un modelo sobre un plano.

En fin, para terminar esta introducción común, añadiré que no he incluido una sección de agradecimientos porque este libro es el resultado de más de veinte años dedicados a impartir clases y dirigir trabajos sobre estos temas en la ETS de Arquitectura de Barcelona. Principalmente en la asignatura optativa “Simulación Visual en Arquitectura por Medios Informáticos” (1994-2014). Pero también en otras asignaturas y cursos que se mencionan en la contraportada y de otros trabajos relacionados con estos temas de los cuales también he sido responsable. Hubiera sido necesario incluir docenas de nombres y seguro que me hubiera dejado alguno. Pero sí quiero dejar constancia de que he aprendido muchísimo de los cientos de estudiantes que he tenido a lo largo de estos años. En algunos casos, particularmente brillantes, porque me

han mostrado alternativas interesantes o me han dado información que no tenía. En otros porque me han hecho ver que lo que explicaba podía explicarse mejor o estaba basado en ambigüedades o en equívocos de los que no era consciente. A todos ellos mi más sincero agradecimiento.

### § § §

La simulación visual de la iluminación podría llevarse a cabo de un modo muy sencillo si, a) pudiéramos crear luces virtuales que distribuyeran la luz como las reales y si, b) contásemos con un sistema de cálculo que computase con exactitud cómo se redistribuye la luz en una escena tras rebotar en todas las superficies que encuentra en su camino y cómo esta redistribución afecta al color y la intensidad de emisión de cada superficie.

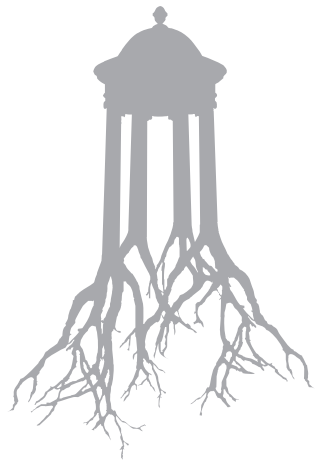
Pero los sistemas actuales no pueden simular con suficiente precisión los complejos modos de distribución de la luz de las fuentes de luz reales y tampoco pueden resolver un sistema de cálculo que requeriría integrar un número casi infinito de rayos lumínicos. Por esta razón, se han desarrollado todo tipo de técnicas: algunas que intentan acercarse a este ideal por métodos de fuerza bruta y otras que utilizan recursos ingeniosos para reducir drásticamente el tiempo de cálculo sin que los resultados difieran demasiado del ideal. Y, como suele ocurrir, todos los métodos que han sobrevivido hasta la fecha, tienen ventajas e inconvenientes que se compensan, de modo que la elección depende inevitablemente de los casos y de las preferencias personales. A lo largo de los seis capítulos del libro, organizados en dos partes, se explican, de un modo que espero resulte suficientemente sintético, estos métodos supervivientes.

La primera parte, *Teoría*, comienza con un análisis de los tres intermediarios de la visión, los objetos, la luz y el sistema visual humano. Cada uno de estos tres intermediarios impone sus límites a lo que percibimos, y comprender de qué naturaleza son estos límites es el primer paso para reproducir o crear escenarios

semejantes a los reales. El capítulo 2 replica este análisis sobre las capacidades y los límites equivalentes para los dispositivos virtuales. El capítulo 3 de esta primera parte proporciona una introducción general a los principales sistemas de cálculo que se utilizan actualmente para simular la iluminación de escenarios virtuales.

La segunda parte, *Técnicas. Análisis de casos*, describe con mayor detalle las herramientas y las configuraciones adecuadas que hay que utilizar en escenarios exteriores, semiexteriores e interiores. Dado que muchas de estas técnicas dependen de recursos materiales sofisticados, tales como tarjetas gráficas o, más exactamente, la “unidad procesadora de gráficos” o GPU (*graphic processing unit*) de que disponga nuestro ordenador, o del uso de formatos especiales como HDR, el capítulo 4 proporciona también una descripción sintética de estos recursos.

Los capítulos 5 y 6 describen con detalle las técnicas que hay que utilizar para preparar la iluminación de un escenario virtual. El capítulo 5 explica los tipos de luces y sus características, así como los objetos y las luminarias virtuales que hay que asociar a estas luces. El capítulo 6 recupera la descripción de los diferentes sistemas de cálculo de iluminación con que se trabaja actualmente y que se han introducido de modo general en el capítulo 3 pero, en este caso, mediante una descripción precisa de las configuraciones utilizadas en alguno de los programas principales que los utilizan y con ejemplos concretos. Los ejemplos están basados principalmente en Mental Ray y en V-Ray pero pueden extrapolarse con facilidad a otras aplicaciones, sobre todo si se han asimilado las ideas básicas que están detrás de los parámetros concretos utilizados, y que pueden variar algo según los casos pero siguiendo los mismos principios.





**TEORÍA**

→ 1





## Escenarios reales

### 1.1 La luz, los objetos y el ojo

En la visión, intervienen tres factores. El primero es la luz, pues no veríamos nada si no hubiera luz. Pero tampoco veríamos nada si no hubiera objetos que captaran la luz, la modularan y nos la devolvieran de diferentes modos. Y es obvio que tampoco veríamos nada si fuéramos ciegos.

En todo lo que sigue, vamos a dejar a un lado casi por completo los objetos. Y no porque no nos importen, sino todo lo contrario. Los colores y las propiedades absorbentes, reflectantes y refractantes de los objetos son tan complejos y tan importantes que requieren otro libro. Otro libro sobre simulación de materiales, que se edita en paralelo a este, como he mencionado en la introducción.

Nos concentraremos, por tanto, en los otros dos factores: las características de la luz y las de nuestro sistema visual, atendiendo principalmente a las características acromáticas, la pura intensidad lumínica y el modo en que esta intensidad se distribuye y se percibe. No creo que haga falta insistir en la importancia de comprender adecuadamente estas características antes de intentar representarlas o simularlas por medios artificiales.

Aunque se aporten datos y valores que es importante conocer, la idea principal que quiere transmitir este primer capítulo es que es mucho más importante ejercitar la observación, mirar y analizar lo que nos rodea y también fotografías, películas, pinturas. La simulación de escenarios solo puede llevarse a cabo adecuadamente si se ha afinado la capacidad de observación.

### 1.2 La luz. Nociones generales. Unidades radiométricas y fotométricas

Desde el punto de vista puramente físico, la luz no es sino energía radiante que nuestro sistema visual puede captar. Más exactamente, es energía radiante comprendida aproximadamente entre los 380 nm y los 780 nm (1 nm, nanómetro, es igual a  $10^{-9}$  metros). El color blanco de la luz es debido a que todas estas energías están actuando simultáneamente y la activación simultánea de todos nuestros receptores se traduce en la sensación que denominamos “blanco”. En este sentido, la luz es una combinación heterogénea de componentes que pueden ser calificados como homogéneos y que, cuando son reflejados por una superficie que absorbe todos los demás, se traducen en la sensación que denominamos “color puro”.

Semejante definición habría dejado boquiabierto a Robert Grosseteste, un *magister scholarum* de la Universidad de Oxford, famoso por los escritos sobre la luz que publicó hacia 1220. Pero no solo porque sería incompatible con su opinión (proveniente de los griegos) de que la visión va del ojo a los objetos y de que el blanco es un color puro que está “más allá” del rojo, así como el negro sería otro color que estaría “más acá” del azul, sino porque la idea de que la luz, “lo más puro”, pudiera ser mezcla de componentes más puros que la propia luz le habría parecido completamente absurda. Lo mismo pensaría Goethe, unos seiscientos años más tarde. Y lo mismo han pensado muchos artistas y filósofos contemporáneos. Y, en cierto sentido, no les falta razón. Pues los razonamientos fenomenológicos no deben confundirse con los razonamientos físicos.



Aunque este libro estará repleto de datos físicos que creo que son saludables, en la medida en que nos previenen de confundir la constitución de lo real con la constitución de nuestras percepciones, quiero volver a insistir en que todos estos datos físicos están subordinados a una intención que busca trascenderlos y tener muy presentes los datos fenomenológicos. La lógica implícita en la apreciación de estos datos inmediatos no coincide ni tiene por qué coincidir con la lógica interna de los fenómenos físicos.

Dicho esto, podemos proseguir resumiendo algunos datos que necesitaremos en adelante.

### Cantidades radiométricas y fotométricas

Si la luz es *energía radiante*, el primer dato que deberemos tener presente es la intensidad de energía radiante de las principales fuentes de luz que vamos a manejar. Y el segundo dato, el modo en que esta energía radiante es captada por nuestro sistema visual, lo cual está directamente relacionado con el modo en que se distribuye esta energía en la región visible.

La energía radiante se mide corrientemente en vatios (julios/segundo), una unidad familiar pues estamos acostumbrados a comprar, por ejemplo, una bombilla “de 100 vatios”. Antes de ver la relación entre cantidades radiométricas y fotométricas, recordemos cómo se aplican. Y el ejemplo de la bombilla “de 100 vatios” es muy adecuado. Como veremos en los capítulos que siguen, en muchos contextos es preferible hablar de cantidades radiométricas, porque así no necesitamos preocuparnos por el modo preciso en que estas cantidades se traducen a cantidades fotométricas según la eficiencia de la luminaria o la curva de distribución de eficacia luminosa en función de la longitud de onda.

La *potencia radiante*, medida en vatios, es una especificación global del flujo radiante de la fuente. Si queremos saber la intensidad de la radiación, necesitaremos dividirla por la unidad de superficie considerada y, en este caso hablaremos de *exitancia radiante*. Como la luz se propaga en todas direcciones a partir de un punto, será más relevante en la mayoría de los casos considerar un ángulo sólido; y en ese caso, hablaremos de *intensidad radiante*. Si lo que nos interesa es la radiación que llega a un determinado punto, lo cual dependerá no solo del flujo radiante o la intensidad sino de la distancia y la orientación, hablaremos de *irradiancia*, que es el flujo radiante recibido por unidad de área. Y, si lo que nos interesa es saber la cantidad de radiación que llega desde un objeto emisor y en una dirección determinada, hablaremos de *radiancia*. Todas estas cantidades radiométricas se relacionan directamente con las correspondientes fotométricas por lo que repetiré estas definiciones con mayor extensión en el apartado siguiente.

### Las cuatro unidades fotométricas principales

Las cantidades radiométricas no nos dicen nada, desde el punto de vista de la visión, si no sabemos cómo se convierten estas unidades radiantes en unidades fotométricas. Si tenemos una fuente de luz de 100 vatios, que emite en la región de los infrarrojos o de los ultravioletas, en una habitación que no cuente con ninguna otra fuente de luz, no veremos absolutamente nada, pues esta energía no entra dentro del espectro visible. Por el contrario, si la fuente de luz fuera un objeto ideal (un cuerpo negro que transforma en luz toda la energía que recibe) cada vatio se converti-

Cantidades radiométricas		Cantidades fotométricas	
Potencia radiante	w (vatio)	Potencia luminosa (flujo)	lm (lumen)
Exitancia radiante	w/m <sup>2</sup>	Exitancia luminosa	lm/m <sup>2</sup>
Irradiancia	w/m <sup>2</sup>	Illuminancia	lm/m <sup>2</sup> (lux)
Intensidad radiante	w/sr	Intensidad luminosa	lm/sr (cd)
Radiancia	w/m <sup>2</sup> sr	Luminancia	cd/m <sup>2</sup>

Figura 1.1 Cantidades radiométricas y fotométricas.



ría en 683 lúmenes, la primera unidad de flujo luminoso que hay que anotar. Y, por tanto, la fuente emitiría 68.500 lúmenes. Una bombilla real tiene un rendimiento mucho menor, por debajo de los 20 lúmenes/vatio. Y su flujo sería, por tanto, no superior a 2.000 lúmenes. Podemos razonar a la inversa para tener una cierta idea de lo que es “un lumen” y llegaremos a la conclusión, por una simple regla de tres, de que “un lumen”, sería la luz emitida por una bombilla de algo menos de 0,5 vatios.

La intensidad de una luz en el origen se puede medir igualmente en candelas, lo que resulta más intuitivo si recordamos que una candela se definió originalmente como la luz emitida por una vela (de tamaño, peso y otras características especificadas con precisión). Esta definición, que se remonta al siglo XIX, se ha sustituido actualmente por otra más precisa (en 1948, la IX Conferencia General de Pesos y Medidas adoptó la definición de *candela decimal* como «la intensidad luminosa emitida por un orificio de  $1/60 \text{ cm}^2$ , irradiando como un cuerpo negro a la temperatura de solidificación del platino medida en dirección perpendicular al plano del orificio») pero nos basta con la idea más o menos vaga, pero fácil de entender, de que una candela es la intensidad que emite una vela. Y el lumen se definió originalmente como el flujo emitido

por una candela sobre un estereorradián. Y el estereorradián es el equivalente volumétrico del radián: una porción de superficie esférica determinada por un cono proyectado desde una esfera de 1 metro de radio.

Por otro lado, si prescindimos del origen, la intensidad de la luz aparente, sean cuáles sean su distancia y su dirección, es otro valor que importará tener en cuenta, y este valor se mide en luminancias. Y, desde el punto de vista práctico, nos interesará saber cuánta luz llega, por ejemplo, a la mesa en que vamos a trabajar. Y esto se mide en luxes. Resumiendo, precisemos las cuatro principales unidades que he introducido en los párrafos anteriores:

a) El *flujo luminoso* es el flujo radiante de una fuente de energía, multiplicado por un valor que traduce la radiación a valores que dependen del rendimiento de la fuente y de la sensibilidad del ojo humano al tipo de radiación. Su unidad es el *lumen*. El valor máximo, ideal, para una fuente que transformara toda su energía en luz, sería de 685 lúmenes por vatio. Las lámparas de mayor eficacia, como las lámparas de sodio a baja presión utilizadas en autopistas, alcanzan los 200 lm/W mientras que las corrientes están entre 10 y 20 lm/W. Una lámpara de 100 vatios emite algo más de 1.000 lm. El flujo es una medida

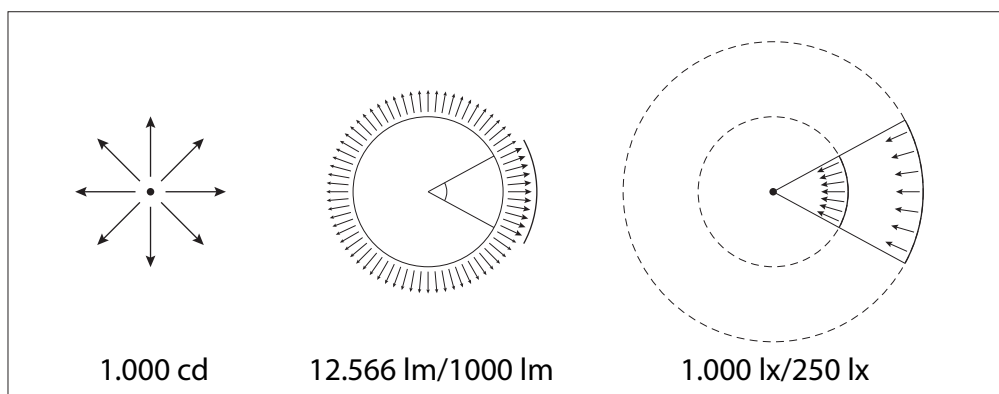


Figura 1.2 Las tres primeras unidades fotométricas principales. 1.000 cd irradiando en todas direcciones (izquierda) suponen 12.566 lúmenes ( $1 \times 4\pi$ , área de una esfera de 1 m de radio) y 1.000 lúmenes de flujo radiante en el área comprendida por 1 estereorradián. La iluminación recibida por esta área sería de 1.000 luxes, que disminuirían con el cuadrado de la distancia.

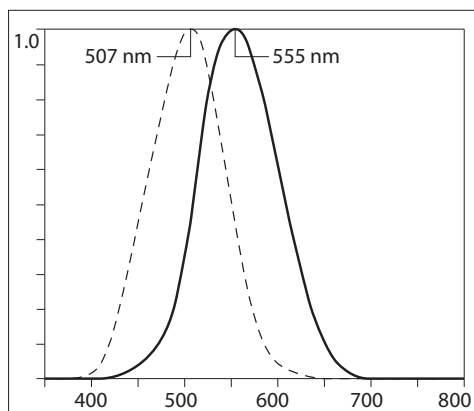


Figura 1.3 Funciones  $V(\lambda)$  de eficiencia luminosa espectral de la CIE (1924, 1951). En trazo continuo, la función fotópica, en discontinuo, la escotópica. Las intensidades se han normalizado para facilitar la comparación.

independiente de la dirección, que es variable en el caso de lámparas concretas.

b) La segunda unidad mide la *intensidad* en una dirección determinada lo que implica que se debe conocer esta distribución a partir de unos diagramas proporcionados por los fabricantes, denominados diagramas goniométricos o diagramas polares. La intensidad se mide en *candelas*. La intensidad de una candela se definió originalmente como la intensidad emitida por una vela "corriente" y la definición actual, adoptada en 1948 y citada más arriba, es más sofisticada pero menos intuitiva. Pero es sencillo relacionar candelas con lúmenes pues si la fuente de luz es 1 lumen y consideramos que la emisión es uniforme, en torno a una esfera de 1 m de radio, la intensidad por metro cuadrado será de  $1 / 12.56$  (la superficie de la esfera es  $4 \times \pi \times r^2$ , es decir, en este caso,  $4 \times 3.1415...$ ). O bien, a la inversa, si la intensidad es de 100 cd, esto

será equivalente, si la distribución es uniforme, a  $\sim 1.256$  lm. Y, si la emisión no es uniforme, tendremos que recurrir a gráficos o tablas que nos den la intensidad para cada dirección de emisión.

c) La tercera unidad, luxes, mide la *iluminación*, que es la que más interesa en las fases iniciales de un proyecto de iluminación. Depende de la fuente, pero también de la distancia, y especifica la cantidad de luz que debe recibir una superficie determinada. Un lux se define como la iluminación de una superficie de  $1 \text{ m}^2$  que recibe un flujo luminoso de 1 lumen. Es una unidad más adecuada si lo que nos importa es saber cuánta luz recibe una superficie, con independencia de cuál sea la fuente de iluminación. Como referencias útiles, se puede recordar que de 1 a 10 luxes es la iluminación que tienen los pasillos sin demasiada luz; de 20 a 40, las calles con alumbrado exterior, de noche; de 100, los pasillos relativamente bien iluminados; de 200, los vestíbulos, habitaciones de poco uso, salas de museos (de 150 a 200 para pinturas al óleo o temple, de 50 a 75 para acuarelas), pistas de tenis y deportes similares; de 300 es lo que suele requerirse para trabajo con requerimientos visuales bajos tales como aulas o salas de trabajo que no requieran gran atención visual; de 400, para pistas de deporte de competición; de 500, para trabajo de oficina con requisitos visuales normales; de 1.000 a 2.000, para trabajos con requerimientos visuales especiales (dibujos de precisión, trabajos de artesanía, laboratorios). La iluminación sobre una superficie se relaciona con la intensidad en el origen por la ley cuadrática inversa que estipula que esta intensidad es inversamente proporcional al cuadrado de la distancia.

lm/W	Fuente
12-17	Bombilla incandescente de tungsteno
14-20	Bombilla halógena
30-70	Tubo fluorescente
90-100	Lámpara de vapor de sodio a baja presión
120-140	Lámpara de vapor de sodio a alta presión
683	Cuerpo negro ideal

Figura 1.4 Valores de eficacia luminosa de diversas fuentes.



d) La cuarta unidad mide la *luminancia*, la intensidad aparente de una superficie o una fuente de luz, que depende de la orientación y se mide en candelas por metro cuadrado,  $\text{cd/m}^2$ . La luminancia es un valor de gran importancia teórica y práctica, y lo primero que hay que recordar es que no depende de la distancia: depende de los rayos que se transmiten en línea recta desde cualquier dirección, por lo cual no es aplicable la ley cuadrática inversa. La luminancia que nos llega a lo lejos en una noche oscura es la misma tanto si su origen es una humilde linterna como una estrella de millones de vatios de potencia. En la literatura anglosajona, se encuentran a menudo los valores de luminancia expresados

en mililamberts,  $\text{mL}$ . Para pasar de  $\text{mL}$  a  $\text{cd/m}^2$  (nits) hay que multiplicar por 3.183. Otra unidad que dejó de utilizarse en 1978 pero puede encontrarse todavía en algunos textos técnicos es el apostilb (asb), que se define como la cantidad de lúmenes por unidad de área. Su relación con las unidades actuales es  $1 \text{ cd/m}^2 = \pi \text{ asb}$ .

Normalmente, se trabaja con dos unidades de referencia, la intensidad y la iluminación. La elección de uno u otro sistema de especificación dependerá de si partimos de una fuente de luz determinada o de unas condiciones de iluminación que queremos conseguir. Como sabe cualquiera que haya proyectado la iluminación de un interior, hay que ir alter-

<i>Luxes</i>	<i>Situación</i>
120.000	Luz solar muy brillante
110.000	Luz solar brillante
20.000	Cielo azul despejado, mediodía
25.000-10.000	Cielo cubierto, mediodía
2.000-1.000	Luz artificial para trabajos de precisión
1.000-500	Luz artificial para trabajos con cierto grado de exigencia
500-200	Luz artificial para trabajos corrientes
400	Amanecer o atardecer, cielo despejado
100	Luz artificial en zonas de circulación
40	Amanecer o atardecer, cielo cubierto
10	Luz artificial en calles
<1	Amanecer o atardecer, cielo muy cubierto
0,25	Luna llena, noche despejada
0,01	Un cuarto de luna, noche despejada
0,002	Cielo estrellado, despejado, sin luna, incluyendo resplandor del cielo
0,0002	Cielo estrellado, despejado, sin luna, excluyendo resplandor del cielo
0,0001	Cielo estrellado, cubierto, sin luna,

Figura 1.5 Valores de referencia generales para iluminancias.

<i>Luxes</i>			<i>Actividad</i>
20	30	50	Áreas de circulación externas
50	100	150	Áreas de circulación internas
100	150	200	Escaleras, almacenes
200	300	500	Trabajo mecánico, montaje de maquinaria pesada
300	500	750	Trabajo intermedio, ensamblaje de elementos grandes
500	750	1000	Iluminación general de aulas
			Trabajo con cierto nivel de exigencia, ensamblaje de elementos pequeños
			Oficinas abiertas
1.000	1.500	2.000	Elaboración de planos, despachos de arquitectura o ingeniería
			Trabajo con alto nivel de exigencia
			Reproducción de color e impresión
1.500	2.000	3.000	Calibración y análisis de elementos pequeños
5.000	10.000	15.000	Trabajo con muy alto nivel de exigencia
10.000	15.000	20.000	Trabajo prolongado con muy alto nivel de exigencia
			Salas de operaciones

Figura 1.6 Niveles de iluminación mínimos, medios y máximos, recomendados para diferentes tipos de actividades. Selección adaptada de la norma ISO 8995.



nando entre estos dos modos de analizar y proyectar la iluminación de un recinto. Por tanto, basta recordar que hay dos tipos de mediciones que nos interesan en la práctica: la que da valores en el origen y la que da valores en el destino. Los valores en el origen se miden en lúmenes (flujo radiante total, con independencia de la dirección) o en candelas/m<sup>2</sup> (intensidad radiante en una dirección determinada). Los valores en el destino se miden en luxes. Así, si tenemos una luz que emite 1 lumen en todas direcciones, una esfera de 1 m de radio recibiría 1 lux por m<sup>2</sup>. Y una esfera de 2 m de radio recibiría el mismo flujo, pero la iluminación sería cuatro veces menor, como es fácil de comprender si nos imaginamos en volumen el esquema que se muestra en la figura 1.2 y que resume estas tres unidades principales.

Estas cuatro unidades se corresponden con las unidades radiométricas, flujo radiante (flujo luminoso, lúmenes), intensidad radiante (intensidad lumínica, candelas), irradiancia (iluminación o iluminancia, luxes) y radiancia (luminancia, cd/m<sup>2</sup>).

También podemos resumir todo lo anterior con un ejemplo simple. Supongamos que tenemos una fuente de luz esférica, de 25 cm de radio con una potencia de 100 vatios y un rendimiento luminoso de 20 lm/vatios, y queremos hallar su flujo luminoso, su intensidad y la iluminación que proporcionará a un plano situado a 2 m de distancia. Para simplificar el ejemplo, supondremos que emite uniformemente en todas direcciones.

El flujo luminoso se obtiene directamente de su factor de rendimiento, es decir:

$$\Phi = 100 \text{ W} \times 20 \text{ lm/W} = 2.000 \text{ lm}$$

La intensidad se obtendrá dividiendo el flujo por la superficie de emisión de una esfera unitaria cuyo ángulo sólido es  $4\pi$ , es decir:

$$I = \Phi / \omega = 2.000 / 4\pi = 159 \text{ cd}$$

La iluminación se obtendrá dividiendo esta intensidad por el cuadrado de la distancia (si el punto considerado no estuviera enfrente la distancia, sería mayor y la iluminación, menor), es decir:

$$E = I / d^2 = 159 / 4 = 39,8 \text{ lx}$$

cd/m <sup>2</sup>	Fuente
3.000.000.000	Cuerpo negro a 6.500 °K (> 10 <sup>9</sup> )
2.300.000.000	Superficie del sol, con radiancia media de $2.3 \times 10^7 \text{ W.m}^{-2}.\text{sr}^{-1}$
1.600.000.000	Superficie del sol al mediodía (desde la tierra, con radiancia media $1.4 \times 10^7 \text{ W.m}^{-2}.\text{sr}^{-1}$ )
200.000.000	Lámpara de mercurio, alta presión
20.000.000	Lámpara halógena
600.000	Superficie del sol en el horizonte (desde la tierra, con radiancia media $1.4 \times 10^7 \text{ W.m}^{-2}.\text{sr}^{-1}$ )
120.000	Lámpara incandescente, 60 W
50.000	Máxima tolerancia visual (resplandor)
10.000	Lámpara fluorescente (hasta 14.000)
10.000	Llama de vela, fuego de chimenea
10.000	Nube blanca iluminada
8.000-7.000	Cielo despejado ( <i>clear sky</i> )
2.500	Luna llena
2.000-1.500	Cielo cubierto ( <i>overcast sky</i> )
500-50	Luminarias visibles, no molestas, en espacios interiores
300-100	Monitor plano (zona blanca de luminosidad máxima)
110	Superficie de 0,7 de reflectancia iluminada con 500 luxes
100	Papel blanco, de 80 % de reflectancia (iluminado con unos 400 luxes)
50	Papel gris, de 40 % de reflectancia (misma iluminación)
5	Papel negro, de 4 % de reflectancia (misma iluminación)
~0,01	Umbral de visión de color en visión nocturna
~0,00001	Umbral corriente del ojo adaptado a visión nocturna
~0,00001	Cielo estrellado, despejado, sin Luna, excluyendo resplandor del cielo
~0,000003	Umbral de visión (10 <sup>-6</sup> )
~0,000001	Cielo estrellado, cubierto, sin Luna

Figura 1.7 Valores de referencia para luminancias (adaptado de IESA Lighting Handbook y otras fuentes).



La luminancia se obtiene dividiendo la intensidad por el área proyectada que, en este caso, será una circunferencia de 0,25 m de radio, es decir:

$$L = 159/\pi r^2 = 810 \text{ cd/m}^2$$

### **Luminancia, radiancia, irradiancia, reflectancia, brillo**

En la literatura científica, se encuentran a menudo varios términos o conceptos estrechamente relacionados y que pueden dar lugar a confusiones. Este apartado intentaré aclarar estos conceptos aunque esto implique alguna repetición de lo que ya he resumido en los apartados anteriores.

Cuando hablamos de *intensidad* de una fuente, estamos propiamente hablando de *radiancia*, de energía radiante: la cantidad de energía transferida por unidad de área (expresada corrientemente por unidades tales como W/m<sup>2</sup>). Y las diferencias son lineales, es decir, 200 W/m<sup>2</sup> de intensidad es el doble que 100 W/m<sup>2</sup>. Volveremos a esto en el capítulo siguiente pues los voltajes que se aplican a un monitor controlan la intensidad que necesitamos regular. Por otro lado, en muchos trabajos sobre simulación de la iluminación, se prefiere, el término radiancia porque es más objetivo y no requiere precisar la longitud de onda (el color) que haría variar el término fotométrico correspondiente.

Cuando hablamos de iluminación, estamos propiamente hablando de *irradiancia*, de la energía radiante que llega a una determinada superficie y que se mide en las mismas unidades que la anterior. En este caso, lo que varía es el concepto. Una misma superficie puede ser considerada como emisora, y en este caso hablaríamos de radiancia, o como receptora, y en este caso hablaríamos de irradiancia. Como en el caso anterior, la mayoría de los trabajos sobre simulación de la iluminación prefieren este término, porque es más objetivo.

Cuando hablamos de *reflectancia* estamos hablando de un valor que depende de la superficie y es también independiente de nues-

tra percepción. Pero este valor se traduce en luminancias y el sistema de visión humano tiene la notable capacidad de distinguir aquellos valores de luminancia que son debidos a la iluminación global de una escena, de aquellos que son debidos a las características de una superficie, manteniendo la relación entre reflectancias y “descontando” la iluminación. En una misma escena, el contraste entre una superficie blanca y una negra rara vez sobrepasa el rango 30:1. Los rangos mucho mayores que se dan en cualquier escena real se deben a la iluminación. Pero la relación entre reflectancias se mantiene: una superficie negra a la luz del día puede emitir mucha más luz que una superficie blanca de noche. Pero, en ambos casos, las percibimos como negra y blanca, pues lo que percibimos es la relación de reflectancias en un entorno con una iluminación global determinada.

Cuando hablamos de *luminancia* estamos hablando también de un valor que depende de la fuente y de su intensidad y que es independiente de nuestra percepción concreta. Pero está traducido a valores que se corresponden con lo que podemos percibir a partir de una función normalizada, que se resume en la curva mostrada en la figura 1.3. La luminancia es la iluminación aparente de una fuente de luz, tal como la recibimos. Este “tal como la recibimos” quiere decir que se tiene en cuenta el modo en que el ángulo de observación modifica la intensidad emitida por la fuente original. La luminancia es un concepto que se aplica a superficies aparentes, tanto si son emisoras de luz como si la reflejan.

En el caso de una superficie emisora, la luminancia no decrecería con la distancia, pues este decrecimiento es debido a que consideramos una misma cantidad de iluminación que se reparte sobre un área cada vez mayor. Pero, en el cálculo técnico de la luminancia, se considera un haz luminoso limitado a un sector diferencial que puede ser tan pequeño como sea necesario. Concretamente, en este caso, la luminancia se define por la fórmula

$$L = d\Phi / dA \times \cos\theta \times \Omega$$





en donde  $L$  es la luminancia,  $d\Phi$  es el flujo luminoso transmitido por un rayo elemental en la dirección considerada,  $dA$  es el área de la sección de ese rayo elemental,  $\theta$  es el ángulo abarcado por el rayo y  $\Omega$  es el ángulo sólido correspondiente al rayo.

En el caso de una superficie reflectante, la luminancia depende de tres factores principales: la iluminación que recibe la superficie, las propiedades reflectantes de la superficie y la posición del observador con respecto a la superficie. En este caso, la luminancia se define por la fórmula

$$L = E \times \xi / \pi$$

en donde  $E$  es la iluminancia en luxes que recibe la superficie y  $\xi$  es su reflectancia.

La luminancia se puede medir, por tanto, de dos modos, a partir de la fuente emisora o a partir de la superficie reflectante. Si una fuente de luz tiene una intensidad de 1 cd y un área de 1 m<sup>2</sup>, su luminancia es de 1 cd/m<sup>2</sup>. Si una superficie es difusora perfecta y su reflectancia es del 100 % (dos condiciones ideales que no se pueden dar en la realidad) y recibe 1 lux de iluminación, su luminancia es también de 1 cd/m<sup>2</sup>. Puede entenderse este último caso como homologable al primero, si se considera la superficie como una luz secundaria. El ejemplo más claro de este modo de ver las cosas es la Luna, que es una fuente de luz secundaria bien conocida. Otro ejemplo importante de luz secundaria es la bóveda celeste. Para los casos que se analizarán más tarde, conviene recordar que la luminancia del cielo despejado (*clear sky*) está en torno a las 8.000 o 7.000 cd/m<sup>2</sup> y la del cielo cubierto (*overcast sky*) está en torno a las 2.000 o 1.500 cd/m<sup>2</sup>.

Por otro lado, la luminancia puede confundirse con otros dos términos también utilizados con frecuencia: el “brillo” y la “luminosidad”.

Cuando hablamos de *brillo* (*brightness*) estamos hablando de un atributo característico de la percepción visual según el cual un área determinada nos *parece* que emite una cantidad mayor o menor de luz, con indepen-

dencia de que esto sea realmente así o de las causas de que esto sea así. Por tanto, no es una cantidad que se pueda medir de un modo objetivo, aunque es posible *relacionarla* con mediciones objetivas.

Cuando hablamos de *luminosidad* (*lightness*) estamos hablando de algo muy parecido a “brillo” pero que no es exactamente lo mismo, si aceptamos la definición de la Commission Internationale de l’Éclairage (CIE). Este organismo define la “luminosidad” como un atributo característico de la percepción visual según el cual un área nos parece que emite una cantidad mayor o menor de luz *en relación con otra área que se percibe como “blanca”*. Por consiguiente, es un atributo que implica la percepción de estímulos relacionados. Por esta razón, también se le ha denominado a veces “brillo relativo”. La CIE define la luminosidad (*lightness*) por medio de la fórmula:

$$L^* = 116 (Y/Y_n)^{1/3}; \text{ con } Y/Y_n > 0.008856$$

en donde  $Y$  es la intensidad del estímulo e  $Y_n$  es la intensidad de referencia “blanca”.

Los valores en que se basan estas definiciones asumen que la luminosidad sigue una curva aproximadamente logarítmica, un tema importante que retomaremos más adelante.

### 1.3 Tipos de luces

La primera distinción, algo más escurridiza de lo que parece, en relación con los “tipos de luces”, es la que cabe hacer entre luces naturales y artificiales. Esta distinción se remite a los tipos de objetos emisores, y las fronteras entre un emisor “natural” o “artificial” son difusas.

En cualquier caso, resulta evidente que la principal fuente de luz “natural” es, para nosotros, la luz diurna. Así que comenzaremos por aquí.

Los cálculos de la contribución de la luz diurna a la iluminación interna de los edificios están estrechamente relacionados con la simulación visual de la iluminación. Esta relación es directa, en la mayoría de los casos, pues se trata de reproducir con exactitud el





modo en que la luz diurna penetra en los interiores o ilumina los exteriores.

En los análisis de la luz diurna, se utilizan dos métodos principales: los que buscan calcular la iluminación absoluta, en luxes (o *foot-candles*), y los que determinan la iluminación relativa.

En el primer caso, se utilizan modelos complejos tales como el *lumen input method* o el *flux transfer method*, que permiten hacer predicciones relativamente precisas sobre la iluminación que llegaría a un determinado punto de un recinto. En el segundo caso, se utiliza principalmente un indicador que no emplearé en los cálculos de simulación pero que merece la pena recordar: el “factor de luz diurna” o DF (*daylight factor*). Este concepto fue introducido en 1895 por A. P. Trotter y desarrollado por J. M. Waldram, entre otros, entre 1920 y 1940, aproximadamente. Y ha sido y sigue siendo un concepto popular, a pesar de sus limitaciones, pues utiliza porcentajes en lugar de medidas absolutas. Se define como la proporción entre la iluminación recibida por una superficie horizontal interior ( $E_{in}$ ) y la que recibiría una superficie horizontal ( $E_{ext}$ ) exterior sin obstáculos, en las mismas condiciones. Se considera que la distribución celeste es conocida en términos relativos (en relación con el cenit de un cielo de luminancia media).

Por ejemplo, para  $E_{in} = 250$  lx y  $E_{ext} = 8100$  lx el *daylight factor* sería:

$$DF = 100 * E_{in} / E_{ext} = 100 * 250 / 8100 = 3,09 \%$$

Estos valores, así obtenidos, proporcionan referencias tales como  $DF < 2 \%$  (el interior es oscuro y requeriría luz artificial todo el día);  $DF: 2 \%$  a  $5 \%$  (el interior se ve iluminado y hay un buen equilibrio entre la temperatura y la iluminación);  $DF > 5 \%$  (la iluminación natural es muy fuerte y habría problemas potenciales de temperatura, sobre todo en verano), que resultan útiles en la práctica profesional pues dan valores relativos y fácilmente comprensibles y porque se subdividen en tres componentes (el Sol queda excluido): el componente celeste (SC, *sky component*), el componente reflectante externo (ERC, *exter-*

*nal reflected component*) y el componente reflectante interno (IRC, *internal reflected component*). Cada componente se calcula por separado por medio de tablas, diagramas o fórmulas, y el resultado se suma para obtener el DF. Véase Robbins (1986, cap. 8 y cap. 9) o Pritchard, 1999, cap. 6) para más información sobre estos métodos.

## El Sol. Radiación solar

La principal luz natural es, para nosotros, el Sol por tanto, los primeros datos que nos importan son los correspondientes al Sol.

La superficie radiante del Sol tiene una temperatura media de unos 5.800 °K. La radiación es mayor en la zona de ondas medias, correspondiente a los 500 nm y disminuye hacia las ondas cortas y algo menos hacia las largas. La cantidad total de energía emitida por la superficie del Sol disminuye con el cuadrado de la distancia (unos 149.5 millones de kilómetros de media entre el Sol y la Tierra) y, al llegar a la Tierra, la irradiancia directa normal (o perpendicular a los rayos solares) fuera de la atmósfera, que recibe el nombre de constante solar, tiene un valor medio de 1.366 W/m<sup>2</sup> (que corresponde a un valor máximo en el perihelio de 1.395 W/m<sup>2</sup> y un valor mínimo en el afelio de 1.308 W/m<sup>2</sup>). Estas cifras se corresponden con los datos recogidos por satélites, si bien estos datos son muy variables, no solo porque la radiación del Sol no es constante, sino por dificultades en las mediciones. En bastantes casos, se utiliza el dato de 1.200 W/m<sup>2</sup>. Pueden encontrarse datos más precisos por internet si se busca con las palabras clave “*solar irradiance data*” y “NASA” pues la NASA ha promovido muchas misiones para obtener datos sobre la irradiación solar que están disponibles para el público en general.

De esta cantidad, solo un 40 % aproximadamente llega a la superficie de la Tierra. Y alrededor de un 30 % (el albedo o reflectividad media de la superficie terrestre) se refleja y es devuelta al exterior.

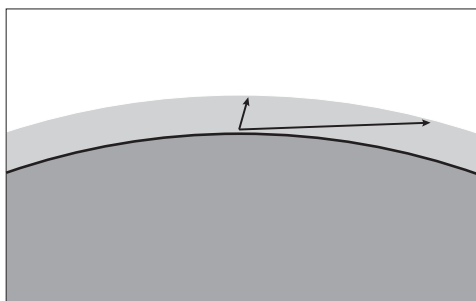


Figura 1.8 Espesor de la atmósfera que los rayos del Sol deben atravesar cuando está cerca del cenit y cuando está cerca del horizonte.

En términos fotométricos, el Sol tiene un rendimiento luminoso de unos 93 lúmenes por vatio (incluyendo infrarrojos y ultravioletas). Para calcular la iluminación que llega a un punto, se utiliza una fórmula que depende de varios factores (la fecha, la hora y determinados parámetros que dependen de la posición) y de la constante solar. A partir del valor anterior, la iluminación media en el exterior de la atmósfera sería del orden de los 127.000 luxes ( $1.367 \times 93$ ). Los valores que se obtienen por la aplicación de la fórmula mencionada y teniendo en cuenta los factores de atenuación de la atmósfera, etc., dan resultados en torno a los 100.000 luxes, aunque las variaciones son considerables. Por ejemplo, en Barcelona, a las 12.00 del mediodía, a principios de julio, es del orden de 99.000 luxes y, a la misma hora de principios de enero, del orden de 81.000.

La luminancia del Sol visto desde la Tierra es del orden de los  $10^9$  cd/m<sup>2</sup>. Es el valor más alto que podemos observar en circunstancias corrientes, a sabiendas de que mirar directamente el Sol durante varios minutos puede ocasionar daños irreparables en la retina.

Los valores de iluminación y luminancia del Sol se han incluido en las tablas adjuntas.

La iluminación dada por el Sol sobre una superficie horizontal es máxima al mediodía, cuando el Sol está en el cenit, y mayor en verano que en invierno, cuando los rayos son más perpendiculares. La iluminación sobre una superficie vertical es, en principio, mayor

cuanto más cerca está del horizonte. Sin embargo, el espesor y la densidad de la atmósfera también aumentan al acercarse al horizonte, tal como puede apreciarse en la figura adjunta. De ahí que la iluminación máxima no se dé cuando está cerca del horizonte, sino cuando la altitud del Sol (medida sobre el horizonte) es de un ángulo mayor, que puede ser del orden de unos 30°.

La tabla adjunta proporciona datos sobre la iluminación en la dirección perpendicular a los rayos del Sol. La masa de aire es una medida de la longitud relativa del recorrido del Sol en la atmósfera que se obtiene, en la mayoría de los casos, por la  $\text{cosec}(h)$ . Esta tabla está basada en estudios realizados por P.Moon en 1940 y en mediciones de laboratorios ingleses, y recogidos en la obra citada de Hopkinson *et al* (1966) y que se aplica a climas soleados (como España), en la columna c1, y a países menos soleados (como Inglaterra), en la columna c2, con valores del orden de un 70 % inferiores. La reducción debida a la masa de aire puede obtenerse mediante las fórmulas que se dan en el manual mencionado y otros ( $\log_{10} E = 4.1 - 0.1 \text{cosec}(h)$  para países soleados y  $\log_{10} E = 3.95 - 0.1 \text{cosec}(h)$  para países menos soleados).

Todos estos valores están revisados, actualizados e incorporados a tablas internas

$h$	$m$	$c1(lx)$	$c2(lx)$
5°	10,40	12.912	9.038
10°	5,60	38.306	26.792
15°	3,80	56.813	42.502
20°	2,90	70.155	49.066
30°	2,00	86.080	60.256
40°	1,55	95.226	66.604
50°	1,30	100.714	70.586
60°	1,15	104.049	72.845
70°	1,05	106.201	74.352
80°	1,02	107.170	75.105
90°	1,00	107.600	75.320

Figura 1.9 Valores medios de iluminación perpendicular a los rayos del Sol (adaptado de Moon/Hopkinson *et al* (1940/1966) en función de la altitud ( $h$ ) en grados y la masa de aire ( $m$ ) en climas soleados ( $c1$ , cielos más despejados) y países menos soleados ( $c2$ , cielos menos despejados).



incluidas en programas informáticos de simulación, de manera que, afortunadamente, no tenemos que preocuparnos por utilizar fórmulas o consultar tablas. Pero no está de más recordar que son resultados que vienen de lejos y que tienen en cuenta factores que a menudo no se toman en consideración, como la variación debida a la masa de aire según la dirección de los rayos del Sol.

## El cielo. Radiación celeste

La segunda fuente de luz natural que nos importa es la bóveda celeste. La iluminación diurna está filtrada y distribuida por la atmósfera, que actúa como una gran pantalla que suaviza y hace más homogénea la luz del Sol. Cuando las nubes ocultan el Sol resulta más evidente este papel de gran pantalla de la atmósfera. La irradiación global de la atmósfera depende de un fenómeno denominado *dispersión* (*scattering*, en la literatura anglosajona) que se analizará más abajo.

La radiación del cielo cambia con las condiciones atmosféricas, la estación del año y la hora del día. Y la distribución de la radiancia celeste (*sky radiance*) varía con el ángulo azimutal y con el ángulo de incidencia. Y es más alta en la dirección del Sol. Pero también varía con la longitud de onda. Para valores inferiores a los 500 nm, es más homogénea y se concentra en torno al Sol. Para longitudes de onda más largas es más heterógena y aumenta cerca del horizonte en la dirección del Sol. Estas variaciones son debidas a los efectos de dispersión mencionados, que varían con la longitud de onda. Pero también tienen que ver con el albedo, pues se ha comprobado que este aumento cerca del horizonte es mayor en las regiones antárticas donde el albedo es superior a 0,90. De hecho la distribución de la radiancia celeste parece corresponderse significativamente con la distribución del albedo.

Todo esto hace muy difícil crear un modelo preciso, pues se necesita una función que tenga en cuenta múltiples parámetros. Sin embargo, es necesario contar con algún tipo de

modelo para diversos tipos de aplicaciones, incluidos los sistemas de simulación visual de la iluminación. Y calibrar adecuadamente el grado de simplificación de cada modelo. Esto se ha hecho en varios casos mediante la toma de datos reales y su comparación con los datos derivados de modelos teóricos. Como es obvio, los datos reales varían considerablemente en función del lugar, la fecha y la hora, y resulta difícil generalizarlos.

El análisis de la distribución de luminancias en el cielo se ha desarrollado desde hace mucho tiempo y se ha intensificado aún más en los últimos años debido al interés creciente por desarrollar métodos que permitan aprovechar mejor las fuentes de energía natural, entre otras causas. Desde el punto de vista de la simulación de iluminación de exteriores, puede decirse que, en la actualidad, hay cuatro referencias principales, con las variantes que indicaré en cada caso: la distribución uniforme, la distribución propia de un cielo uniformemente cubierto, la distribución propia de un cielo uniformemente despejado y la distribución propia de casos intermedios que, de algún modo, pueden encerrarse en una descripción sintética.

Este análisis se remonta a los trabajos pioneros de Pierre Bouguer (1698-1758), un gran astrónomo y matemático francés. El análisis científico de la luz diurna y su influencia en los edificios comenzó a finales del siglo XIX. Los primeros métodos, como los diagramas de Waldram (de 1923) o los de Pleijel (de 1954) o Turner (de 1969) eran principalmente gráficos. En paralelo, se desarrollaron análisis precisos de la distribución de luminancias en el cielo, como los que llevaron a cabo pioneros como Basquin en Chicago, publicados en 1906 y, sobre todo, Kimball y otros colaboradores en Chicago y Washington, en el período 1921 - 1923 que dieron lugar a algunos de los primeros modelos de distribuciones para cielos cubiertos (1923) y despejados (1929). En Europa, los primeros trabajos importantes de recopilación de datos tuvieron lugar en Teddington (en el suroeste de Londres), entre 1923 y 1929, seguidos por muchos otros en Alemania, Francia, Finlandia o Austria, entre otros países.



En 1929, Pokrowski propuso una nueva fórmula para calcular la distribución de luminancias en un cielo despejado basándose en las fórmulas de Rayleigh para el cálculo de la dispersión. Posteriormente, se han llevado a cabo muchas mediciones de la distribución de luminancias en un cielo despejado en muchas partes del mundo. Estas mediciones han confirmado que la fórmula general dada por Pokrowski se cumple con bastante exactitud.

En 1942, Moon y Spencer recopilaron y reordenaron las principales fórmulas y trabajos previos, y propusieron otra fórmula empírica para calcular la distribución de luminancias en un cielo cubierto. Y, posteriormente, en 1955 y 1967, Kettler propuso otra fórmula para un cielo despejado.

En 1955, la CIE adoptó la fórmula de Moon y Spencer para el cómputo de la distribución de luminancias de su modelo de cielo cubierto (*overcast sky*) y, en 1965 y 1973, la fórmula de Kettler para el cómputo de cielo despejado (*clear sky*). Bastantes años más tarde, en 1993, Richard Perez y otros colaboradores propusieron un modelo que cubría todos los casos, denominado en algunos programas de simulación “Perez-All-Weather”, a partir de medidas de irradiancia que se integraban en un modelo general que abarcaba desde condiciones despejadas hasta condiciones cubiertas, pasando por condiciones de nubosidad variable.

Se han propuesto muchos otros modelos para cubrir mejor los casos intermedios entre los dos extremos principales para los modelos no uniformes. Véase Kittler *et al* (2012) para los avances más recientes. Pero los únicos modelos disponibles por ahora en los programas de simulación, como veremos más adelante, son los citados. El resto de esta sección se dedica a resumir las ideas básicas en que se han basado estos modelos.

### § § §

a) La primera referencia que he mencionado y la primera que se utilizó fue la de una distribu-

ción de luminancias *uniforme*, prescindiendo de la luz solar. En estas condiciones, la iluminación de un punto de una superficie horizontal rodeada sin obstáculos por el hemisferio celeste cubierto y sin influencia del Sol viene dada por la fórmula

$$E = \pi L$$

con  $E$  en luxes y  $L$  en  $\text{cd/m}^2$ .

En el Reino Unido el National Physical Laboratory ha tomado varias mediciones de la iluminación sobre el suelo (véase tablas en Pritchard, 1999, p. 126). En estas condiciones, los valores medios que se acordaron fueron de 5.000 lx (aunque los valores máximos llegaban, a las 12 h de julio, a 35.000). Valores similares, en torno a los 5.500 lx (~5 % de la correspondiente al Sol) se han adoptado en otros países del norte de Europa. Un valor medio de 5.500 lx correspondería, según la fórmula anterior, a 1.750  $\text{cd/m}^2$ .

La iluminación recibida en la Tierra, debida al Sol, en un país soleado, es del orden de 98.000 a 108.000 lx. Por tanto, el suelo, cuya reflectancia media puede ser del orden de un 10 % tendrá una luminancia de unos 10.000 asb que equivalen ( $1/\pi$ ) a unos 3.200  $\text{cd/m}^2$ . Con ángulos menores, la iluminación directa del Sol se atenúa cada vez más por la atmósfera. A unos 20°, puede ser del orden de los 70.000 lx. Una fachada con un 10% de reflectancia media recibiría por tanto unos 7.000 asb equivalentes a unos 2.200  $\text{cd/m}^2$ .

### § § §

b) La segunda referencia mencionada es la de una distribución de luminancias *no uniforme para el cielo cubierto*, estandarizada según las fórmulas adoptadas por la CIE, para el *CIE Standard Overcast Sky*.

Este modelo se asemejaría a un cielo oscuro, cubierto con gruesas nubes, y es el primer modelo con una distribución de luminancias no uniforme.

En este caso, la distribución depende del ángulo según la fórmula:



$$\frac{L_{\gamma}}{L_z} = \frac{1+2 \operatorname{sen} \gamma}{3} = \frac{1+2 \cos Z}{3}$$

donde:

- $L_{\gamma}$  es la luminancia celeste ( $\text{cd/m}^2$ ) del elemento considerado.
- $L_z$  es la luminancia del cenit.
- $\gamma$  es el ángulo del elemento considerado sobre el horizonte.
- $z$  es la distancia angular entre el elemento considerado y el cenit ( $Z = 90^\circ - \gamma$ ).

Esta fórmula está basada en la propuesta por Moon y Spencer, citada más arriba. Aplicando esta fórmula a un caso práctico nos encontramos con que la distribución de luminancias sigue un patrón regular, con el cenit más luminoso y un decrecimiento regular de la luminancia hacia el horizonte en donde alcanza un valor que es aproximadamente 1/3 del valor del cenit. Los valores en el cenit pueden ser del orden de  $1.500 \text{ cd/m}^2$  y en la zona de los  $10^\circ$  del orden de los 500.

Aplicando la fórmula de Moon y Spencer se pueden obtener valores de luminancia correspondientes a una determinada altura sobre el horizonte ( $h^\circ$ ) multiplicando el valor de referencia uniforme por el factor  $Z$ , siendo el factor  $Z$  el valor correspondiente a la luminancia media de esa altura, obtenida mediante la aplicación de dicha fórmula. La

$h$	$Z$
5	0,50
10	0,58
20	0,72
30	0,86
40	0,98
42	<b>1,00</b>
50	1,08
60	1,17
70	1,23
80	1,27
90	1,29

Figura 1.10 Valores relativos de luminancia del cielo cubierto con respecto a la altura sobre el horizonte en grados ( $h$ ) y respecto a la luminancia media ( $L_m$ ). CIE Standard Overcast Sky. Valores basados en la fórmula de Moon y Spencer.

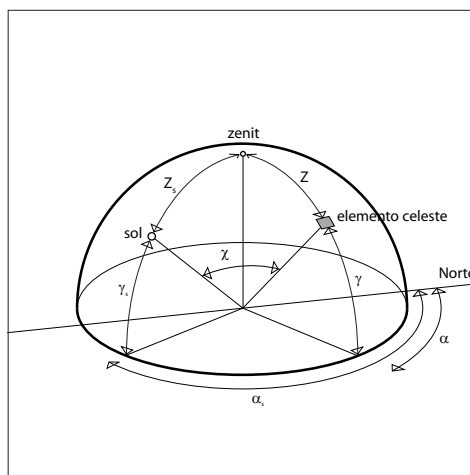


Figura 1.11 Ángulos y direcciones sobre la bóveda celeste que definen la posición del Sol y de elementos del cielo en el modelo de cielo despejado de la CIE (de Kittler y Darula, 2002).

tabla adjunta de la figura 1.10 muestra los valores relativos resultantes.

Esta fórmula ha sido corregida posteriormente por varios investigadores (Fritz y Petherbridge, pocos años después) para incluir el efecto de reflexión de la Tierra, que tiene un efecto considerable en los ángulos bajos, tanto más cuanto más reflectante es la Tierra, como ocurre en las zonas nevadas, donde el valor de reflectancia normal (en torno al 10 %) aumenta considerablemente (hasta 80 % o algo más).

§ § §

c) La tercera referencia mencionada correspondería a una distribución de luminancias *no uniforme para un cielo despejado*, estandarizada según las fórmulas adoptadas por la CIE, para el *CIE Standard Clear Sky*.

Este modelo se asemejaría al cielo de un día soleado, con luminancias que varían con la altitud y el azimut, mayores en torno al Sol y menores en la zona opuesta.

En este caso, la fórmula, que fue adoptada en la ISO/CIE 1996 es la siguiente:



$$L_{\gamma\alpha} = L_z \frac{(1 - e^{-0,32/\sin\gamma})(0,91 + 10e^{-3\chi} + 0,45\cos^2\chi)}{0,274 (0,91 + 10e^{-3Z_s} + 0,45\cos^2 Z_s)}$$

donde:

- $L_{\gamma\alpha}$  es la luminancia del elemento de cielo considerado.
- $L_z$  es la luminancia del cenit.
- $\chi$  es la distancia angular entre el elemento de cielo y el Sol (ver figura adjunta).
- $Z_s$  es la distancia del cenit al Sol.

La primera aportación a este caso, ya citada, fue debida a Pokrowski (en 1929). La fórmula de Pokrowski daba el valor de la luminancia para un punto, en la dirección correspondiente a un ángulo determinado sobre el horizonte, en función de tres parámetros principales: la distancia angular del punto considerado con respecto al Sol, un coeficiente de dispersión primario  $P$  (con un valor aproximado de 0,32) y un coeficiente de dispersión secundario  $K$  (con un valor de 0 a 5, según el contenido de polvo y humedad, si bien en la mayoría de los casos se utilizaba un valor de  $K = 0$ ).

Aplicando esta fórmula a un caso práctico, observamos que aparece una zona brillante alrededor del Sol, con una luminancia de cerca de 30.000 cd/m<sup>2</sup> y una zona de azul oscuro de más de 1.000 cd/m<sup>2</sup> enfrente del Sol, a unos 90° en la dirección opuesta. Tanto la zona brillante como la azul oscura se mueven con el Sol a lo largo del día. La luminan-

cia de la zona azul será menor en atmósferas claras y a mayores alturas, aunque es difícil apreciar directamente y se vería más claramente si se utiliza un espejo cóncavo.

La fórmula de Pokrowski fue revisada y refinada por Kittler (en 1955, 1976 y 1998) y tras ser adoptada por la CIE, es la que se utiliza en la actualidad. Los términos básicos son similares a los de Pokrowski pero con una medición más precisa de la dispersión en torno al Sol, principalmente, y con una mejor adecuación a los datos experimentales.

En el artículo de Kittler y Darula (2002) citado en referencias, puede encontrarse más información y ejemplos de cálculo.

En Hopkinson et al (1966) se pueden encontrar tablas, basadas en fórmulas y mediciones diversas, a partir de trabajos de varios investigadores, que proporcionan valores medios de luminancias en diferentes partes del mundo, en diferentes periodos del año y en diferentes zonas de la bóveda celeste.

### §§§

d) La cuarta distribución mencionada de luminancias para un caso intermedio, es decir para un *cielo parcialmente cubierto*, es obviamente más complicada, pues las variaciones son demasiado grandes. Pese a esto se han probado diferentes fórmulas que permitan obtener una estimación media.

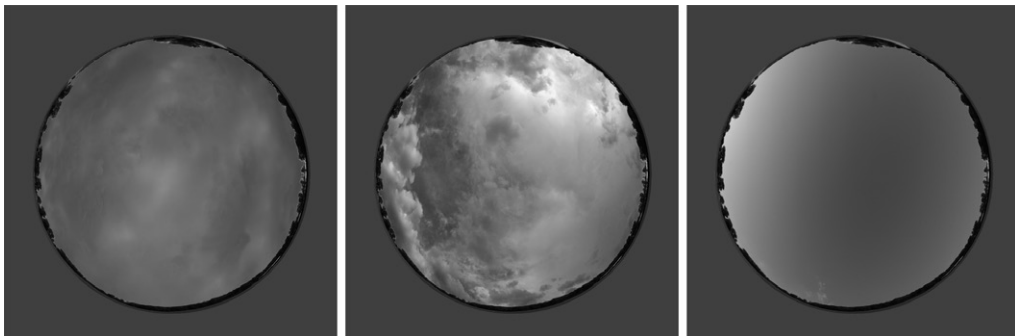


Figura 1.12 Ejemplos de imágenes que corresponderían a cielo cubierto, intermedio y claro captados con un objetivo de 180°.





A lo largo de los últimos años se han propuesto distintos modelos.

Pleijel demostró que las variaciones medias son también proporcionales a la altitud del Sol y que podía utilizarse una fórmula que permitía, de un modo aproximado, computar la iluminación media. La fórmula de Pleijel, modificada (Hopkinson *et al*, *op cit*, p 47) sería:

$$E_H = 500 h$$

donde  $E_H$  sería la iluminación media para un cielo parcialmente cubierto correspondiente a una determinada altura  $h$  en grados. Pero esta aproximación es mucho más débil para valores bajos de  $h$  porque debido a la dispersión de la luz en la atmósfera, hay una iluminación crepuscular del orden de 1.000 lx, muy variable según la latitud.

Entre los primeros modelos intermedios, se puede citar el propuesto por Nakamura *et al* (en 1985); el modelo propuesto por Littlefair (1994) denominado ASRC-CIE que es una combinación de cuatro modelos: el cielo despejado y cubierto estándar de la CIE, una versión del cielo despejado estándar muy turbido (propuesto por Gusev) y una fórmula intermedia, propuesta por Nakamura (1985). Otro modelo intermedio fue propuesto por Igawa en 1997 a partir de veinte patrones de distribución de luminancias basadas en mediciones tomadas en Tokio. Kittler *et al* propusieron otro modelo en 1998, basado en mediciones realizadas en Tokio, Berkeley y Sydney, que fue propuesto a la CIE e incorporado a una propuesta provisional en 2001. Otros modelos propuestos en esta época, hacia 1998, también basados en varios coeficientes, fueron los de Perradeau y colaboradores. El modelo propuesto por Preetham *et al* (1999) es similar al modelo de Perez, pero utiliza un parámetro especial, denominado "turbidez" (*turbidity*) para elaborar un modelo analítico que proporciona valores absolutos de luminancia y cromaticidad para el cenit y coeficientes de distribución de luminancia y cromaticidad.

El inconveniente principal de estos métodos es que se basan en datos estadísticos

que solo son válidos para unos pocos países. Una alternativa es generar imágenes de cielos particulares con escáneres celestes a partir de las cuales puedan computarse las luminancias. Pero el precio de estos escáneres es prohibitivo. Otra alternativa, más barata, es intentar medir las luminancias a partir de imágenes tomadas con una cámara digital adecuadamente calibrada y un ordenador que procese los datos. Pero este procedimiento, que sería sin duda el más atractivo, resulta aún demasiado complejo (hay que tomar un gran número de imágenes) y los resultados no son extrapolables fácilmente. Roy *et al* han publicado trabajos hacia 1998 con algunos intentos pioneros en esta línea. Por otro lado, las mediciones excesivamente locales tampoco resultan prácticas, por razones obvias.

El modelo más utilizado y que proporciona buenos resultados es el propuesto por Richard Perez y otros colaboradores (Perez *et al*, 1991, 1993), elaborado a partir del análisis de mediciones realizadas en Berkeley. Este modelo se basa en cinco parámetros relacionados con la obscuridad o brillo del horizonte, el gradiente de luminancia cerca del horizonte, la intensidad relativa de la región cercana al Sol y la dispersión relativa de la luz. Utilizando los mismos términos que en las fórmulas anteriores para facilitar la com-

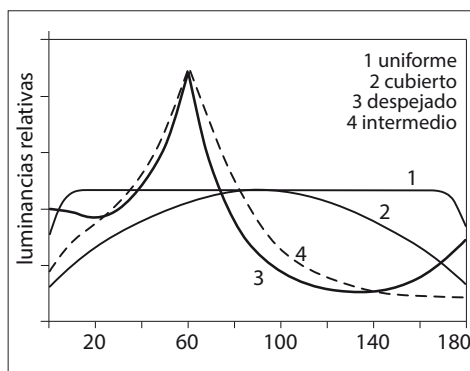


Figura 1.13 Distribución de la luminancia celeste para 4 modelos CIE, generada con el programa Radiance para el sol a 60° de altitud y a lo largo de un hemisferio que pasa por la posición del Sol.



octante	estación	0°-30°	30°-60°
cara al sol	mitad verano	5200 - 8500	5200 - 8500
	equinoccios	3400 - 8200	3400 - 7500
	mitad invierno	3400 - 7900	2000 - 3800
opuesto al sol	mitad verano	1400 - 5200	1400 - 2400
	equinoccios	1400 - 2800	1200 - 1550
	mitad invierno	1200 - 1550	1000 - 1400
90° al sol	mitad verano	1700 - 5200	2000 - 3800
	equinoccios	1700 - 3800	1700 - 2900
	mitad invierno	1700 - 2900	1200 - 1900

Figura 1.14 Valores medios de luminancia (cd/m<sup>2</sup>) del cielo despejado. Adaptado de Hopkins et al (valores redondeados).

paración, el modelo de Perez vendría dado por la fórmula:

$$F(Z, \chi) = (1 + Ae^{B/\cos Z})(1 + Ce^{D\chi} + E \cos^2 \chi)$$

de la cual se deriva la luminancia  $L_p$ :

$$L_v = L_{vz} \frac{F(Z, \chi)}{F(0, Z_s)}$$

y donde  $Z$  es el ángulo correspondiente a la altura del elemento celeste considerado;  $\chi$  el ángulo entre el Sol y el elemento de cielo considerado; y  $Z_s$  el ángulo correspondiente al Sol. Los coeficientes representan la distribución de la luminancia debida a diferentes factores:

- oscuridad o brillo del horizonte,
- gradiente de luminancia cerca del horizonte,
- intensidad relativa de la región en torno al Sol,
- anchura de la región en torno al Sol,
- retrodispersión relativa de la luz.

En otro modelo posterior (elaborado por Kittler, R. y Perez hacia 1998), se modeló la distribución de luminancias por medio de funciones especiales de gradiente (*gradation and indicatrix functions*), con las cuales se definieron quince tipos de cielos que irían de “despejado” a “cubierto”.

Todas las referencias dadas hasta aquí no pretenden otra cosa que dar indicaciones sobre el contexto en que han surgido algunos de los principales modelos de simulación de la luz diurna que utilizaremos en los capítulos

correspondientes más adelante. El lector interesado en profundizar en estos temas puede consultar la bibliografía que se da en el capítulo de referencias, como los textos clásicos de Hopkins et al (1966) o Robbins (1986) o buscar más información por internet.

## El fenómeno de la dispersión

La intensidad y la calidad de luz celeste, la luz que irradia de la atmósfera, están relacionadas directamente con el fenómeno general de la *dispersión* (*scattering*, en la literatura científica anglosajona), la desviación de la trayectoria rectilínea de la radiación por irregularidades del medio que atraviesa. Por tanto, es un fenómeno propio de la interacción de la luz con los materiales y, por esta razón se trata más extensamente en el libro citado al comienzo sobre materiales. Pero, en la medida en que afecta directamente a la calidad de la luz, hay que resumir aquí al menos lo principal.

Los fenómenos de dispersión tienen lugar en la atmósfera terrestre, principalmente en las primeras capas. La atmósfera terrestre se considera compuesta de cuatro capas: troposfera (hasta los 10 o 12 km de media), estratosfera (hasta los 50 km, aproximadamente), mesosfera (hasta los 80 o 90 km) y termoesfera (hasta más de 500 km). La densidad y la presión atmosférica varían con la altura y dependen de muchos factores. Los fenómenos climáticos tienen lugar en la troposfera, que tiene una extensión variable hasta un mínimo de unos 6 km en las zonas





polares y hasta un máximo de unos 20 km en el Ecuador.

La atmósfera contiene un gran número de gases (78 % de nitrógeno, 21 % de oxígeno) y partículas de orígenes diversos, vapor de agua y polvo. Las partículas absorben energía a diferentes longitudes de onda, según sus propiedades específicas. Además, las partículas dispersan la energía solar. Esta dispersión puede tener lugar una o varias veces, por lo cual hay que distinguir, a la hora de analizar este fenómeno, entre la dispersión primaria, secundaria y múltiple. La figura 1.15 muestra, de un modo esquemático, este proceso que llega a la superficie de la Tierra desde múltiples direcciones.

El análisis de la dispersión es complejo y debe tener en cuenta, entre otros aspectos, los recorridos ópticos, que dependen de lo que se denomina *masa óptica*, que depende a su vez de las variaciones de la propia superficie de la Tierra y de la densidad del medio. Aquí no entraremos en estos detalles, sino que resumiré tan solo las ideas básicas que pueden tener repercusión en las técnicas de simulación.

La dispersión depende fundamentalmente del tamaño y del índice de refracción de las partículas. Las variaciones se pueden analizar en casos característicos, como la dispersión de la luz por el humo, la niebla o el propio aire.

El fenómeno de la dispersión se puede apreciar más claramente si observamos el humo. Si se observa el humo de una chimenea contra un cielo oscuro, su color es azulado. Las partículas que forman el humo dispersan la luz azulada más que los otros componentes de la luz. Si el color del cielo es azul claro, el color es anaranjado o amarillento. Esto es debido a que los componentes de la luz incidente ya se han dispersado contribuyendo al color del cielo y vemos la dispersión de los componentes restantes, que son de mayor tamaño.

Pero hay muchas variaciones que dependen del tamaño de las partículas. Cuando las partículas no son mayores de 1/10 o 1/5 de la

longitud de onda media de la luz (entre 500 y 600 nm), la dispersión es proporcional a  $1/\lambda^4$  y crece con rapidez hacia el lado de las ondas cortas, azules y violetas. Es decir, las ondas cortas, por debajo de los 400 nm (azules y violetas) se dispersan diez veces más que las largas.

Sin embargo, con partículas mayores, la dispersión es proporcional a  $1/\lambda^2$  y esta desviación hacia las ondas cortas es mucho menos pronunciada. El humo de un cigarrillo que se consume en el cenicero es azulado porque las partículas son muy pequeñas. El humo exhalado por el fumador tiene un tono más cálido, menos azulado. Esto es debido a que, después de haber sido inhaladas y exhaladas, quedan recubiertas por una capa de vapor de agua que aumenta su tamaño. Lo mismo ocurre con el fenómeno citado del humo de las chimeneas, cuyas partículas son de tamaño mayor y tienen un color tirando a marrón oscuro.

El color azulado del cielo es debido al mismo fenómeno. En verano, es menos azul porque aumenta el número de partículas de arena o polvo, de mayor tamaño. Pero, en días despejados, cuando el viento o la lluvia han barrido estas partículas, es de un azul más intenso, pues son las minúsculas partículas, las propias moléculas del aire, las que dispersan la luz hacia las ondas cortas del espectro.

Los primeros modelos que explicaron adecuadamente la dispersión de la luz por efecto de partículas minúsculas fueron propuestos por el físico inglés Lord Rayleigh (1842-1919)

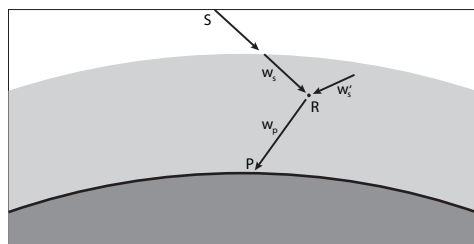


Figura 1.15 Esquema de las interacciones y variables que tienen lugar con la dispersión. P es el punto al cual llega la luz desde el sol, S, después de ser dispersada por el punto R, que recibe tanto rayos directos ( $w_s$ , dispersión primaria) como indirectos ( $w'_s$ , dispersión secundaria).



Figura 1.16 *Dispersión Rayleigh (Rayleigh scattering) provocada en todas direcciones por las moléculas del aire, más pequeñas que la longitud de onda media de la luz.*



Figura 1.17 *Dispersión Mie (Mie scattering) provocada en direcciones determinadas por las moléculas de aerosoles, polvo, gotas de humedad, iguales o mayores que la longitud de onda media de la luz.*

y sintetizados en la ley que lleva su nombre y cuyo enunciado es:

$$s = k (n - 1)^2 / N \lambda^4$$

donde  $s$  representa la cantidad de dispersión por unidad de volumen,  $n$  es el índice de refracción,  $N$  es el número de partículas por  $\text{cm}^3$  y  $\lambda$  es la longitud de onda. Hay muchas otras versiones de esta fórmula, que dependen del tipo de análisis.

Cuando las partículas son mayores, aproximadamente iguales a la longitud de onda, la forma de las partículas tiene mayor influencia y la descripción física del fenómeno requiere otras formulaciones, que fueron dadas por el físico alemán Gustav Mie (1869-1957), por lo cual se conoce como “dispersión de Mie” (*Mie scattering*). La fórmula de la dispersión de Mie es más compleja y también hay múltiples versiones que el lector interesado encontrará sin dificultad por internet o en alguno de los artículos citados en la bibliografía.

La figura 1.15 muestra el esquema general de las interacciones que tienen lugar. La luz del Sol,  $S$ , llega a un punto  $R$  desde la dirección  $w_s$  y se dispersa en una dirección primaria  $w_p$  que llega al punto de observación  $P$ . A lo largo de este recorrido, se va atenuando con lo cual su intensidad alcanza un valor que puede calcularse mediante una fórmula general que tiene en cuenta diversos factores, prin-

cipalmente la distancia recorrida. Por otro lado, al punto  $R$  llegan otros rayos, como  $w'_s$  que provienen de la dispersión que ha tenido lugar en otras partículas. Esta dispersión secundaria también alcanza el punto  $P$ , el cual recibe una suma de intensidades, que pueden estimarse en función de los factores mencionados.

### La perspectiva aérea

Fenómenos similares se manifiestan también en lo que se conoce como *perspectiva aérea*. La disminución de intensidad y el corrimiento de los tonos hacia el azul y el violeta caracterizan la “perspectiva aérea”, claramente presente en la obra de los pintores flamencos del siglo  $\text{xvi}$ , como Patinir, descrito con precisión por Leonardo da Vinci en sus cuadernos y que se popularizó en el siglo  $\text{xix}$ , probablemente a raíz de la publicación de un tratado escrito por el pintor Pierre-Henri de Valenciennes en 1800, *Éléments de perspective pratique*.

Cuanto más lejos están las montañas, mayor es el espesor de aire entre ellas y el observador y mayor es, por tanto, la dispersión. Este gradiente de dispersión nos da claves aproximadas para juzgar la distancia a que se encuentran las diferentes cadenas de montañas o elementos diversos; de ahí la

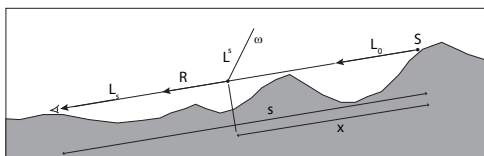


Figura 1.18 Esquema de las variables que intervienen en la perspectiva aérea.

denominación, que asocia estas claves a las de la perspectiva.

Según Minnaert (1954) las líneas de luminosidad constante se pueden trazar fácilmente con un poco de práctica y siguen las "reglas" siguientes. El punto más oscuro se encuentra sobre un círculo que pasa por el Sol y que está a una distancia de unos  $95^\circ$  del Sol cuando está bajo y a una distancia de unos  $65^\circ$  cuando está alto. A través de este punto, pasa una franja oscura que divide el cielo en dos partes, una parte más brillante que rodea al Sol y otra parte brillante en dirección opuesta. La forma y el tamaño de estas franjas varían con la altitud del Sol.

Esta distribución se puede explicar por una combinación de los fenómenos siguientes: a) la iluminación del cielo aumenta en la dirección del Sol; b) a una distancia de  $90^\circ$ , el cielo tiende a hacerse más oscuro, pero esto no acaba de ocurrir así debido al tercer fac-

tor; c) la intensidad de la luz aumenta desde el cenit hacia el horizonte, al mismo tiempo que el color pasa de un tono más azulado a un tono más blanco. Este último efecto es debido a que, en la dirección del horizonte, el espesor de la atmósfera es mayor. El color de las partículas azuladas producto de la dispersión se hace más débil al tener que atravesar una mayor distancia atmosférica.

El color del cielo varía considerablemente de un día para otro. Es más intensamente azul después de que las lluvias o el viento o ambos fenómenos hayan limpiado la atmósfera. Se hace más blanco cuando aparecen nubes de tipo cirrus, y, en verano, cuando hay más partículas de polvo pesadas en la atmósfera, cambia considerablemente en diferentes regiones de la Tierra y en diferentes horas del día en una misma región... Pueden darse explicaciones muy generales, pero es imposible proporcionar explicaciones precisas que cubran todas estas variaciones.

En un día soleado, con el cielo azul, aproximadamente el 80 % de la iluminación global es debida al Sol y el restante 20 % al cielo.

Si tomamos la iluminación media del cielo como 1, la intensidad de una nube blanca sería de 10 y la del Sol 300.000.



Figura 1.19 Imágenes de montañas del Ampurdán solapadas en perspectiva.



Figura 1.20 Luz nocturna.

El efecto de la perspectiva aérea es debido, por tanto, a la dispersión y a la absorción que se produce cuando los rayos de luz atraviesan la atmósfera, junto con la mezcla con otros rayos provenientes de la dispersión de otras partículas. Dado que tanto la dispersión como la absorción varían con la longitud de onda, se produce un desplazamiento espectral hacia las ondas cortas y una pérdida de saturación.

La figura 1.18 resume de forma esquemática las variables que intervienen en este fenómeno y que se utilizan para desarrollar procedimientos de simulación. En esta figura,  $L_o$  representa la radiación de un punto  $S$  de montañas lejanas que es percibido desde el punto  $P$ , situado a una distancia  $s$  y a donde llega una radiación  $L_s$ , que se ha ido atenuando a lo largo de esa distancia. Sin consideramos un punto interior  $R$  de este recorrido, a la distancia  $x$  sobre  $s$ , y denominamos  $L_{in}$  la radiación que emite, esa radiación dependerá de la luz que llega a ese punto por efecto de la dispersión desde múltiples direcciones, una de las cuales,  $L^s(w)$ , será la representada en la figura. La fórmula general para obtener la radiación en el punto que buscamos será:

$$L_s = fL_o + L_{in}$$

donde  $f$  es factor de extinción para la luz que viaja a lo largo del trayecto  $PS$ . Y el problema será hallar el valor de  $L_{in}$ , para lo cual será necesario integrar todas las contribuciones correspondientes a las diferentes direcciones de dispersión en diferentes direcciones  $w$ . El análisis completo de este problema se puede encontrar en Preetham (2003).

Estas variables se han utilizado para elaborar modelos de representación que buscan reproducir este efecto y que se comentarán en el capítulo siguiente. Véanse Hoffman y Preetham (2002) o Dobashi (2002).

## La luz nocturna

La luz de la Luna es luz enviada por el Sol y reflejada por su superficie. Depende, por tanto de la luz emitida por el Sol, de la distancia de la Luna al Sol, del albedo de la superficie lunar y de la posición relativa de Sol, Luna y Tierra. Asumiendo que la fase que nos interesa es la correspondiente a la Luna llena, estos valores son relativamente constantes. Sin embargo, el albedo cambia con rapidez con pequeñas variaciones del ángulo formado en el vértice del triángulo Sol-Luna-Tierra, con valores que van desde el 16 % para un ángulo cercano a los  $0^\circ$  en este vértice y 11 % para un ángulo de unos  $5^\circ$ . Las mayores variaciones de luminancia se dan en función de estas



variaciones de albedo, pero los cálculos y las observaciones coinciden en que la luminancia de la Luna llena está entre los  $2.500 \text{ cd/m}^2$  y los  $4.200 \text{ cd/m}^2$ , si bien el primer valor es el que se da más corrientemente.

El cielo, en general, es luminoso durante la noche. El 50 % de su luminosidad se debe a los millones de estrellas, visibles y no visibles, un 5 %, a la dispersión de luz de las estrellas visibles desde la Tierra y el resto a la propia luminosidad de la Tierra. La intensidad general es muy variable y obedece a múltiples causas, principalmente a variaciones en la emisión terrestre.

Sería posible elaborar modelos similares a los que hemos visto para el cielo pero su utilidad es mucho menor por lo que, aunque es posible que existan modelos, no los he buscado y no los incluyo en este apartado. Baste con algunas observaciones de interés. Durante la noche, el cielo es más claro en el horizonte, con un máximo hacia los  $15^\circ$ ; es una luz que emana de la Tierra, como una especie de débil aurora. Hay también una línea más luminosa que puede verse en determinadas circunstancias y que se denomina a veces “luz del zodiaco”, y que se cree

que se debe principalmente a dos causas: a) un gran disco de polvo cósmico que rodea al Sol, b) la propia luz nocturna dispersada por las capas superiores de la atmósfera. Estos dos efectos dan lugar a una zona piramidal, que varía según las estaciones del año y la posición geográfica.

Aparte de estas observaciones de interés muy relativo, la Luna y las estrellas son una buena referencia para estimar valores de luminancias y diferencias mínimas perceptibles, por lo cual volveremos a este tema más adelante, desde otro punto de vista.

Tampoco está de más, en esa línea, recordar las variaciones de luminancia de un mismo objeto en diferentes condiciones de iluminación.

Si tomamos la intensidad de un objeto negro iluminado por el Sol en un día despejado como 1,00, la intensidad de un objeto blanco sería del orden de 50,00. A la luz de la Luna, el objeto negro tendría una iluminación de 0,005 y el blanco, de 0,25. Es decir, el rango dinámico del objeto a la luz del Sol sería de 50:1 y a la luz de la Luna sería el mismo (0,25/0,005). Sin embargo, en términos absolutos, el “negro” a la luz del Sol sería del orden

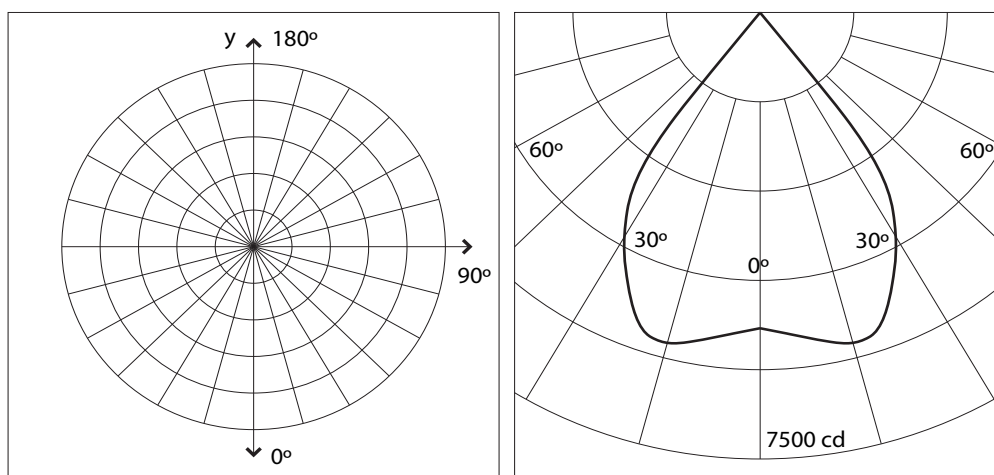


Figura 1.21 Diagramas goniométricos. La figura de la izquierda muestra una sección vertical del diagrama básico con la indicación de los ángulos y las coordenadas polares. Las líneas que irradian del centro corresponden a la dirección de las intensidades (ángulo  $\theta$ ) y las líneas concéntricas, a intervalos de intensidades en cd.

de 4.000 veces más brillante que el “blanco” a la luz de la Luna.

## Luces artificiales

Además de luces naturales tenemos luces artificiales, objetos que emiten luz. La distinción entre “natural” y “artificial” es bastante ambigua, como ya he señalado, pues la incandescencia y la luminiscencia no dejan de ser fenómenos naturales. Una vela emite luz por causas similares a las del Sol pero una bombilla incandescente también. Y hay objetos o animales luminiscentes y lámparas fluorescentes, que emiten luz por la excitación de vapores gaseosos.

Pero podemos aceptar esta distinción si subrayamos otra característica. En el caso de las luces artificiales podemos actuar directamente sobre ellas. Primero, porque las pode-

mos encender y apagar y, segundo, porque podemos regular el modo en que iluminan, sea cambiando su intensidad o su posición, o su dirección o su distribución por medios simples o por medios más complejos, diseñando luminarias con unas características determinadas.

Como la casuística es infinita, me limitaré a resumir las características principales que nos interesa tener presentes en el caso de las luces artificiales.

Un dato importante para simular una luz artificial es conocer cómo se distribuye su intensidad. Para obtener la distribución de intensidades se utilizan datos proporcionados por los fabricantes o generados en un laboratorio por medio de un goniómetro. Un goniómetro es un aparato que incorpora un fotómetro fijado a un soporte móvil, y que puede tomar medidas de una luminaria en diferentes direcciones, por lo general con la ayuda de un espejo que facilita la medición. Así, se obtiene una serie de datos que pueden asociarse a un campo vectorial, en donde cada vector, en una dirección determinada, representaría la intensidad de la luz.

La distribución de intensidades se representa por medio de diagramas goniométricos, que utilizan coordenadas polares. La fuente de luz se sitúa en el centro de una esfera con el eje Y apuntando hacia arriba. La esfera se divide en meridianos y paralelos, de los que resultan dos valores dados por un ángulo vertical,  $\theta$ , y un ángulo horizontal,  $\pi$ . El ángulo  $\theta$  varía de  $0^\circ$  a  $180^\circ$ , con el polo sur en  $0^\circ$ . El ángulo  $\pi$  varía entre  $0^\circ$  y  $360^\circ$ . Si la distribución de intensidades es simétrica, podemos utilizar una sección vertical 2D y prescindir de las variaciones que se proyectarían en horizontal, del ángulo  $\pi$ . A veces, se utiliza una sección vertical 2D indicando con diferentes líneas las dos distribuciones verticales principales, la correspondiente a los ángulos, que irradian del centro y la correspondiente a las intensidades, dadas por círculos concéntricos. En las figuras adjuntas se muestra un caso simple, que res-

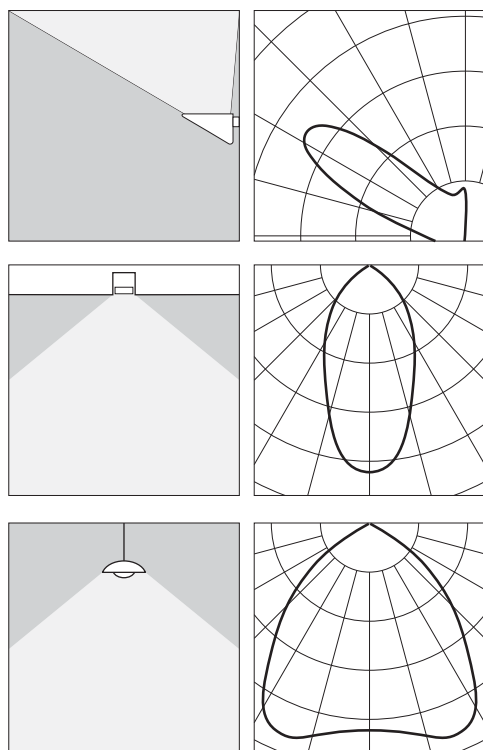


Figura 1.22 Ejemplos de diagramas goniométricos para algunas luminarias corrientes.





ponderaría a una distribución arbitraria, con valores de intensidades señalados en las direcciones principales, y algunos ejemplos de diagramas que podrían corresponder a tipos corrientes de lámparas.

Estos valores también se pueden dar en forma de tablas, con columnas que asocian un ángulo vertical, por lo general en incrementos de 15°, en un rango de 0° a 90°, y con intensidades (cd) correspondientes a cada uno de esos ángulos.

## Lámparas y luminarias

Aunque a menudo tiendan a confundirse, pues los límites que las diferencian son imprecisos, es importante distinguir entre *luces*, *lámparas* y *luminarias*. Una distinción que, como veremos, es aún más importante, por razones técnicas, en el caso de escenarios virtuales.

Una *luz* es un objeto que emite luz. Pero este tipo de objetos están unidos a otros por razones diversas: para que contribuyan a distribuir la luz de un modo más adecuado, para que soporten la luz, para que la protejan de golpes, para que alberguen los cables y mecanismos que regulan su funcionamiento, y, en fin, para que contribuyan al interés y a la calidad visual del entorno (algo que no siempre se consigue).

El término *lámpara*, en castellano, se ha utilizado hasta no hace mucho como equivalente a *luminaria*, mientras que este término se utilizaba a menudo como sinónimo de un cierto tipo de luz o resplandor (las dos primeras acepciones que daba el Diccionario de la Real Academia eran “Luz que se pone en ventanas, balcones, torres y calles en señal de fiesta y regocijo público” y “Luz que arde continuamente en las iglesias delante del Santísimo Sacramento”. Y el diccionario de uso del español de Manuel Seco *et al.*, sitúa, delante de las anteriores, el genérico “luz o resplandor”). Pero, como en otros casos, se ha impuesto, técnicamente, el uso anglosajón en que el término *lámpara* (*lamp*) se utiliza como “fuente de luz” y el de *luminaria* (*lumi-*

*naire*), como el correspondiente al equipamiento. Este es el uso que seguiré.

Una *lámpara* es, por tanto, un dispositivo que produce luz. Y este tipo de dispositivos se pueden clasificar de muchos modos, pero el principal es el que atiende al mecanismo de producción de la propia luz. Y, en este sentido, cabría hablar de cuatro grupos principales: incandescentes (de filamento de tungsteno, principalmente halógenas); de descarga a baja presión (de vapor de mercurio, como las fluorescentes o de vapor de sodio como las utilizadas en el alumbrado público); de descarga a alta presión (de vapor de mercurio o de sodio); y un cuarto grupo, más reciente, que sería el de las lámparas basadas en LED (*light emitting diodes*). Pero habría muchas variantes que no entran en esta agrupación y que pueden encontrarse en un buen manual o un catálogo comercial.

Una *luminaria* es un dispositivo, o conjunto de dispositivos, utilizado con una lámpara y con la finalidad principal de controlar la distribución de la luz.

Para cumplir con esta finalidad principal, el diseño de la luminaria, su forma, es el factor determinante y clasifica las luminarias en múltiples tipos y con diferentes criterios: según las características ópticas (directa, semidirecta, indirecta), según el tipo de lámpara (incandescente, halógena, fluorescente), según el uso (viario, peatonal, industrial, doméstico...), según el tipo de montaje (integrada, exenta), etc. No incluiré aquí ni siquiera los principales, dado que este libro va dirigido principalmente a arquitectos y diseñadores, que están más o menos familiarizados con este tema y que deben ser conscientes de su complejidad y de que para esto haría falta otro libro. Pero volveremos sobre los tipos de luminarias más adelante, y esta clasificación, que puede encontrarse en un buen manual de luminotecnica o en catálogos comerciales, estará implícita.

## Color y calidad lumínica

El color de la luz es mucho más variable de lo que somos conscientes. Sin embargo, este



hecho físico debe ponderarse adecuadamente con el modo en que percibimos la luz, pues en nuestra percepción intervienen diversos fenómenos psicofísicos de considerable importancia como la adaptación y la constancia cromática.

Si entramos en una habitación iluminada con una luz azulada en la que hay un plátano encima de una mesa y tomamos una fotografía con una cámara que no incluya un ajuste adecuado del balance de blancos, esta fotografía, contemplada en condiciones normales nos mostrará un plátano de color azulado. Sin embargo, mientras estamos en la habitación percibiremos el plátano y otros objetos de diferentes colores, con su color “real”. Esto ocurre debido a un fenómeno de adaptación denominado “constancia cromática”. El investigador americano Edwin Land demostró, hacia 1965, que nuestra vista posee la notable capacidad de “descontar” la iluminación ambiental, a partir de varias claves, para captar con bastante precisión la reflectancia propia de los objetos.

Aunque todo lo relativo al color aparece en el libro sobre simulación de materiales citado en la introducción, hay que incluir aquí algo sobre el color de las luces. Los valores, en estos casos, se dan principalmente por referencia a la *temperatura de color*, un concepto que también se explica más extensamente en el libro citado.

Durante las primeras horas del día, cuando sale el sol, la temperatura de color de la luz que irradia está por debajo de los 2.500° K y va aumentando con rapidez hasta alcanzar algo más de 4.000° K. A mediodía alcanza su máximo valor que varía entre los 5.000° K y los 15.000° K de un día de verano sin nubes. Por debajo de los 3.000° K el color de la luz es notoriamente rojizo y por encima de los 10.000° K notoriamente azulado, si bien, por el fenómeno de adaptación al que me he referido, los percibiremos igualmente como “blancos”.

Hay varios colores “blancos” de referencia que se han ido utilizando desde 1931 aproximadamente, fecha en la cual los trabajos de

la CIE comenzaron a utilizarse extensamente en todo el mundo. Actualmente, el más utilizado es el D65, que corresponde a una temperatura de color de 6.500° K que sería la iluminación propia de un día con el cielo despejado.

El color de la luz es uno de los primeros aspectos que debemos tener en cuenta al hablar de calidad de la luz, pues el equilibrio es inestable. Un blanco neutro tiende a desplazarse, en términos de temperatura de color, hacia valores inferiores, más cálidos, o hacia valores superiores, más fríos. Esta coloración no es solo física sino emocional: una escena con una determinada tendencia cromática global aparece, efectivamente, más cálida o más fría, entendiendo estos términos como coloraciones afectivas.

Pero la luz también puede tener otras características que dependen principalmente de los tonos dominantes y del contraste.

Una escena puede ser, para empezar, muy luminosa o muy sombría. Y dentro de cada uno de estos dos extremos, hay grandes variaciones que dependen del contraste y de la dirección de las luces dominantes y de las sombras.

El contraste se percibe intuitivamente como la relación entre las zonas más claras y más oscuras de una escena y la rapidez o suavidad del tránsito entre unas y otras. Para una definición más técnica y más precisa del contraste, véanse los apartados que siguen. Pero desde el punto de vista de composición de la escena, implica una determinada configuración de luces primarias y secundarias. Y las luces secundarias pueden ser tanto luces adicionales como objetos muy claros que actúan como reflectores.

La dirección de la luz y la configuración de las sombras proporcionan claves sobre lo que no es visible. La dirección de la luz nos habla de fuentes de luz que no están presentes directamente pero sí indirectamente. Y la configuración de las sombras nos puede mostrar los contornos de objetos que rodean la escena pero que tampoco son visibles directamente: hojas de árboles, contornos de ob-





jetos o personas, persianas venecianas que se proyectan sobre la escena. Ampliaré estos comentarios en la sección siguiente.

En un escenario virtual, como en un escenario teatral o cinematográfico, lo que rodea el escenario es una tramoya. Pero una tramoya que debe estar bien pensada pues la ausencia de elementos no visibles puede hacer que la escena resulte poco interesante o poco creíble. Una leve sombra en una esquina que sugiere que el espacio continúa por ese lado añade un elemento adicional de interés al resultado.

### 1.4 Luces y sombras. Calidad y función visual de las sombras

Las sombras tienen un papel fundamental en la comprensión visual de una escena. Permiten establecer relaciones entre objetos y hacer que formen parte de un todo coherente. Un objeto sin sombras parece flotar sobre el suelo. Y solo mediante las sombras podemos saber si un ob-

jeto está colgando de una pared o forma parte de ella. Pueden revelar el contorno de un objeto visto de frente y, en consecuencia, dar información sobre su forma. Revelan la presencia de objetos presentes en la escena pero que no son visibles desde un determinado punto de vista. Son un aspecto fundamental de la composición y de la tonalidad emocional de una escena.

Muchas de las características de algunos escenarios reales que hacen que estos nos atraigan o se queden grabados en nuestra memoria, han sido analizadas y sintetizadas magistralmente a lo largo de los últimos seiscientos años de historia de la pintura. La provocativa frase, tantas veces citada, “la naturaleza imita al arte”, condensa esta idea: hemos aprendido a apreciar los paisajes, los objetos cotidianos, gracias a la sabiduría visual de los grandes maestros que han sabido destacar en sus pinturas los rasgos visuales de lo que nos rodea y situarlos en un contexto en el que adquieren un sentido.

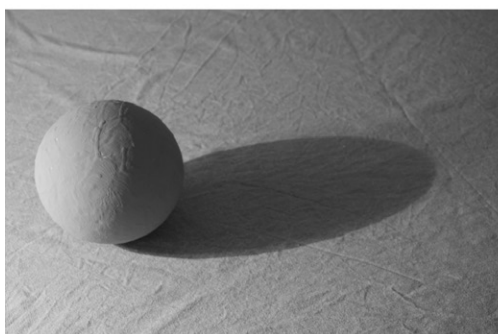


Figura 1.23 Escena con objetos y diferentes tipos de sombras.



Uno de estos rasgos es la ausencia o presencia de sombras. Hasta finales del siglo XVI, hasta Caravaggio, las sombras no comienzan a ser protagonistas de las pinturas. Las grandes obras renacentistas, las de Fra Angelico, Botticelli, Piero de la Francesca, son pinturas luminosas, sin sombras notorias. La aparición posterior de las sombras se ha visto en algunos casos como un símbolo de decadencia y pesimismo, asociado a un gusto malsano por lo siniestro. La propia biografía de Caravaggio, que, entre otras cosas, tuvo que huir de Roma para escapar del arresto por el asesinato de un hombre durante una pelea y, más tarde, de Nápoles por razones similares, ha reforzado seguramente esta interpretación. Le Corbusier clamaba contra Caravaggio y su “nefasta influencia”.

Sin discutir esta interpretación, plausible, me interesa recalcar otra. La pintura que se hacía en la Alta Edad Media, justo antes de la ruptura renacentista, era deudora directa de las grandes vidrieras y los grandes frescos.

No existía la pintura de grandes paneles ni menos aún la de grandes lienzos. La pintura servía para decorar los muros de iglesias y palacios. Y las vidrieras de las iglesias y algunos palacios eran intrínsecamente luminosas: presentaban objetos ideales que irradiaban luz. Objetos que cumplían una doble función: la función simbólica de presentar personajes situados más allá de este mundo, eternamente iluminados, y la función más humana de iluminar, por lo que había que evitar al máximo zonas oscuras que redujeran esta función. Estas dos funciones se mantienen en los grandes frescos, que están pensados para ser iluminados con velas y para destacar de las sombras que envolvían la zona iluminada.

En la época de Caravaggio son cada vez más frecuentes los grandes paneles y los paneles de menor tamaño, que podían estar situados en habitaciones luminosas. Por otra parte, interesa cada vez más el realismo frente al simbolismo. Así se reduce la importancia de las funciones citadas y comienza una indagación que puede decirse que culmina en Rembrandt, pasando por de La Tour o Ribera.

Con Rembrandt aprendimos a apreciar las muchas cosas que pasan dentro de las sombras. No es que, en los que le precedieron, no se encuentren detalles y hallazgos notables. Pero Rembrandt lleva esta indagación a un extremo difícil de sobrepasar. Quien tenga ocasión de contemplar un cuadro real de Rembrandt, no una reproducción, que se acerque (todo lo que le permita el vigilante del museo) a lo que, desde lejos, parece una superficie oscura y uniforme y verá todo un mundo de matices e insinuaciones sutiles.

Si dejamos la pintura y volvemos al mundo real podemos llevar a cabo un simple ejercicio que sirva de introducción a una reflexión sobre cómo las sombras pueden variar por influencia de los objetos cercanos. Las sombras nunca son complementariamente negras: cualquier objeto más o menos cercano orientado hacia las zonas en sombra y que reciba luz, actuará como un reflector secundario que ilumina las zonas en sombra. Una excepción

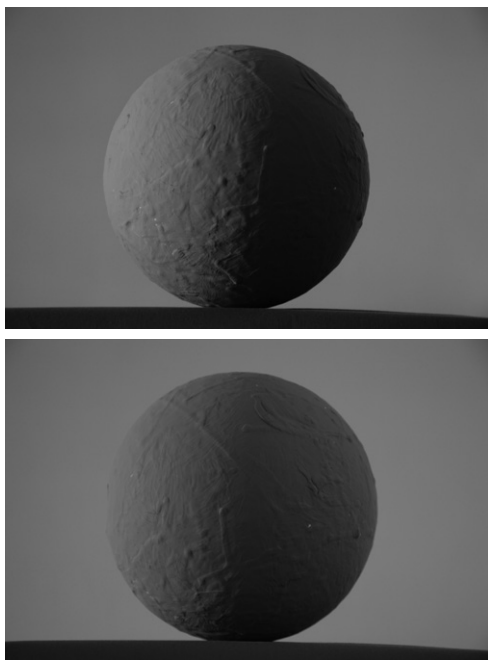


Figura 1.24 Sombras. Objeto esférico con a) un vacío a la derecha, b) una pared a la derecha.



singular, pero muy familiar, es la Luna. Pero si un asteroide de gran tamaño se acercara a la Luna y, en esta situación tan improbable pudiéramos hacer algo, aún más improbable, como fotografiar la cara oculta de la Luna unos milisegundos antes del impacto, lo que veríamos debería ser algo similar a lo que se muestra en la figura 1.24: la cara oculta, iluminada por la luz irradiada desde el lado luminoso del asteroide.

Sin embargo, cuando la Luna nueva acaba de aparecer, justo al inicio del cuarto creciente, se puede ver la cara oculta débilmente iluminada. Esta débil iluminación viene de la propia Tierra que ilumina la Luna. Es una luz cenicienta, bastante variable, a veces claramente visible y otras veces invisible. Estas diferencias se explican porque, en unos casos, en la superficie de la Tierra que se orienta hacia la Luna predominan los océanos (que ocupan casi las tres cuartas partes de la superficie terrestre (71 % para ser exactos) y en otros casos los continentes, así como por las diferencias en nubosidad.

Otro tanto ocurre en situaciones más corrientes como las que se muestran en la figura 1.25 en las que un objeto esférico es ilumi-

nado de diferentes modos por otros objetos cercanos.

Las sombras son también de muy diversos tipos. Hay sombras profundas y muy marcadas, provocadas por luces puntiformes (pequeñas y cercanas o bien grandes y lejanas, como el Sol).

Hay sombras suaves pero nítidas, provocadas por luces como las anteriores pero suavizadas por una luz secundaria muy amplia, o bien provocadas por luces extensas y cercanas.

Hay sombras difusas de infinitos grados de difusión, oscuridad y transparencia, provocadas por luces más o menos extensas, más o menos cercanas, más o menos intensas y que pueden cruzarse entre sí. La iluminación más suave, más difusa, se consigue con luces muy extensas, cruzadas y cercanas al objeto.

El carácter de estas sombras difusas depende, así, de tres factores: el tamaño o extensión de la luz secundaria, la distancia entre la luz y el objeto que emite sombra, y la distancia entre el objeto y su sombra, lo que depende del tamaño del objeto y de su orientación con respecto a la luz.

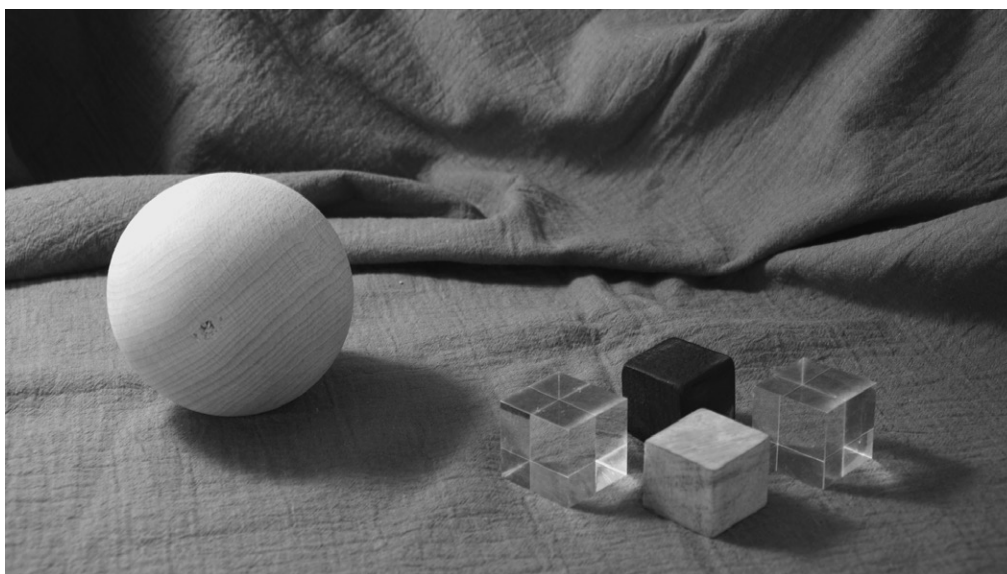


Figura 1.25 Escena con objetos que distribuyen la luz de diferentes modos.



Las sombras pueden contribuir también a singularizar una escena arrojando patrones. Así ocurre en algunos paisajes de Constable que incluyen sombras de nubes. O en escenas en las cuales las sombras de hojas de árboles sobre un camino amplían nuestro campo de visión. O en el interior de una casa en donde aparecen dibujadas con suavidad, sobre las paredes, las sombras de persianas venecianas.

Las sombras tienen diferentes grados de intensidad: hay sombras muy oscuras y sombras muy claras. Pero todas están iluminadas en mayor o menor medida: en el mundo real no existen sombras absolutamente negras pues siempre hay algún objeto que refleja parte de la luz ambiental.

Las sombras proporcionan claves visuales importantes para comprender adecuadamente una escena. Manifiestan la relación que hay entre objetos visibles pero, como ya he apuntado, también pueden revelar la presencia de objetos no visibles. Por otro lado, la iluminación secundaria, necesaria para suavizar las sombras principales, puede arrojar sombras superpuestas que hacen más confusa la imagen. Una de las muchas ventajas, pero peligrosas, de la iluminación virtual es, como veremos, que nos permite utilizar luces que no arrojen sombras.

Las sombras tienen color. Su color viene dado por la iluminación ambiental que las tiñe de tonos más cálidos o más fríos. El color de las sombras arrojadas tiende a teñirse ligeramente del complementario del color de los objetos que arrojan sombras o de la luz principal. En ocasiones, esto es muy notorio, como ocurre con el color de las sombras de objetos sobre la nieve bajo un Sol amarillento, que tienen una tonalidad azulada, un efecto que ya observó y teorizó Goethe.

### 1.5 Los objetos. Modulación de la luz por los objetos

En el apartado sobre tipos de luces podríamos haber incluido los objetos, dado que

otra fuente fundamental de luz es el conjunto formado por las luces secundarias. Este conjunto es imposible de analizar y resumir adecuadamente. Sin embargo, su papel es determinante para caracterizar otra noción fundamental, ya mencionada, pero aún más difícil de concretar: la cualidad lumínica de una escena. Y aquí nos encontramos con otra categoría: los objetos como moduladores de la luz. En cualquier escena real, la calidad de la luz depende, en primer lugar, de las luces principales: el Sol y la atmósfera que envuelve la escena. O, de un modo más general, de las luces primarias y secundarias. Pero, en segundo lugar, depende del modo en que todos los objetos recogen esta luz principal y la redistribuyen de infinitos modos, convirtiéndose a su vez en emisores de luz.

Por razones diversas, este libro es en blanco y negro. No se tomará en consideración ni el color de la luz ni el color de los objetos, un tema tan importante que merece ser tratado exclusivamente en otro libro (en color).

Esto supone excluir una parte fundamental del papel que juegan los objetos o, más exactamente, la superficie de los objetos, en la configuración de escenarios singulares. Los objetos absorben parte de la luz y reflejan otra parte. Y esto se traduce en diferentes colores de variaciones infinitas que no son sino formas diferentes de modular la luz y de desvelar sus componentes armónicos que están entremezclados en la luz blanca y en las sombras.

Pero, aún prescindiendo del color, los objetos modulan la luz de muchas formas que dependen de su reflectancia, de su forma, de las distancias y las orientaciones que mantienen entre ellos.

Todo esto hace de una escena determinada algo singular, irreplicable. Y esto es, sin duda, una obviedad. Pero es una obviedad tan importante que antes de continuar hay que hacerle este espacio, aunque sea pequeño, para subrayar el papel fundamental de los objetos en la distribución de la luz de cualquier escena.

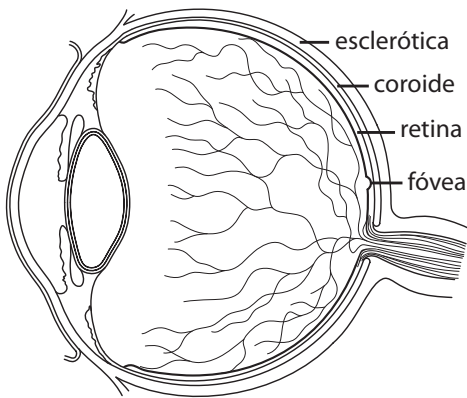


Figura 1.26 Estructura básica del ojo.

## 1.6 El ojo y el sistema visual humano

Esta sección incluye una descripción del sistema visual humano, que es importante para comprender muchas de las características de las imágenes, tanto las que formamos a partir de la percepción de objetos reales como las que se generan sintéticamente. Gran parte de la descripción son inevitablemente iguales a las incluidas en el libro sobre simulación de materiales citado en la introducción. En este caso, sin embargo, se han relegado todos los temas relativos al color, que se tratan más extensamente en el libro sobre materiales y, por el contrario, se ha dado mayor importancia al análisis general de los sistemas de percepción de los valores, a la formación de imágenes, con independencia del color, y al análisis de las características de nuestro sistema visual que afectan a la resolución y el contraste.

### La estructura del ojo

El ojo humano tiene una estructura muy compleja que queda encerrada en una forma simple, una esfera irregular de unos 20 a 24 mm de diámetro. El grosor de esta esfera está constituido por tres capas o envolturas principales que se denominan esclerótica, corioide y retina.

La primera, la *esclerótica*, la más externa, cambia de aspecto y de forma cuando pasa por delante de la pupila, formando la *córnea*, transparente y rodeada de otra zona opaca de color blanco, la esclera, en la cual están insertados los músculos que mueven el ojo en su órbita. La *córnea* se proyecta hacia adelante, con mayor curvatura que la del conjunto del ojo, con un radio de unos 8 mm frente a los 10 o 12 mm del conjunto de la esfera del ojo. La *córnea* contribuye de un modo importante al enfoque, al refractar los rayos de luz que llegan a través del aire. Esto puede apreciarse cuando abrimos los ojos dentro del agua. Los objetos aparecen borrosos debido a que esta contribución de la *córnea* queda anulada, pues su índice de refracción es prácticamente igual al del agua y solo el cristalino contribuye al enfoque. Pero sus capacidades son limitadas, con lo cual la convergencia de los rayos no alcanza a llegar adecuadamente a la superficie de la retina.

La segunda, la *corioide*, recubre internamente la esclerótica y se prolonga detrás de la *córnea* formando el iris, que tiene un color característico y variable para cada persona. El iris incluye una abertura, la pupila, por donde entra la luz. El diámetro de la pupila se adapta a los cambios de luz, haciéndose más pequeño, hasta un límite de unos 2 mm, cuando la luz es muy fuerte y más grande, hasta un límite de unos 8 mm, cuando la luz

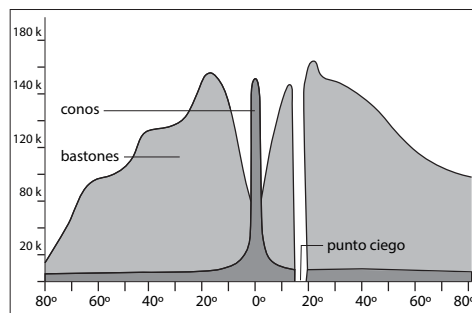


Figura 1.27 Distribución de bastones y conos. El eje vertical indica el número de receptores por  $\text{mm}^2$ . El horizontal, la localización en grados con respecto a la fóvea.



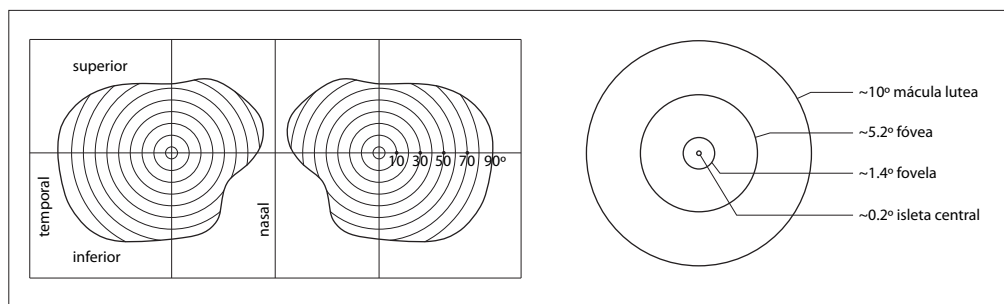


Figura 1.28 Campos de visión principales.

es más débil. Tras la pupila está el cristalino, que funciona como una lente y se alarga o se encoge para variar las características de esta lente y enfocar imágenes cercanas o lejanas. El cristalino está formado por varias capas de diferente índice de refracción, pero cuya media está en torno a 1,42. La variación del grosor del cristalino y de su función como lente, es posible gracias a la acción de los músculos ciliares que, como todos los músculos, se endurecen con la edad y pierden capacidad de acomodación. Entre la córnea y el cristalino hay un medio líquido de índice de refracción similar al del agua, 1,33 que se denomina *humor acuoso*. Y la cavidad general del ojo está llena de un líquido también acuoso y de un índice de refracción similar que se denomina *humor vítreo*.

La tercera envoltura es la *retina*, y es la que incluye los primeros receptores lumínicos de nuestro sistema visual. Estos receptores son de dos tipos y se denominan, por su forma, *bastones* y *conos*.

Los *bastones* solo captan diferencias de luminosidad, son “ciegos al color”. La rodopsina de los bastones no es afectada directamente por la longitud de onda, a diferencia de lo que ocurre con los conos. Sin embargo, esto no quiere decir que no la afecte indirectamente, pues la probabilidad de que un cuanto sea absorbida es mayor para unas longitudes que otras. Alcanza un máximo en torno a los 510 nm y por encima de 800 nm es prácticamente nula. Por otro lado, son mucho más sensibles que los conos a estas diferencias de lumino-

sidad: captan intensidades hasta 1.000 veces menores que los conos.

Las mediciones de la cantidad de luz necesaria para blanquear en un porcentaje significativo la rodopsina fueron desarrolladas principalmente por Rushton hacia 1962 y por Brown y Wald hacia 1964.

Los *conos* captan diferencias de longitud de onda y son de tres tipos lo que, muy aproximadamente, se corresponde con el hecho de que podamos obtener todo tipo de colores combinando tres colores primarios.

El tamaño de bastones y conos es variable según las zonas. Los bastones tienen un tamaño más regular, del orden de las 2 micras en las zonas periféricas, en donde son más abundantes, y entre 1 y 1,5 cerca de la fovea, en la denominada *zona parafoveal*, que abarca alrededor de 8.6°. Los conos tienen de 1 a 4 micras en la región foveal, aunque en varios cálculos se acepta un diámetro medio de unas 2 micras. En la región extrafoveal, el diámetro es mayor, entre 4 y 10 micras, según diversas fuentes. Todos los datos que he resumido hasta aquí, y los que siguen, provienen principalmente de Polyak, Le Grand, y Wyszecki y Stiles (véanse las referencias).

La distribución de bastones y conos es también irregular y se puede describir tomando como referencia lo que podemos denominar la zona central o principal de la retina, porque es la que situamos espontáneamente en el centro de las imágenes que queremos ver. Esta zona central, la *fovea*, es la región de mayor sensibilidad cromática del ojo. La



fóvea tiene alrededor de 1,5 mm de diámetro y abarca un ángulo de  $5,2^\circ$ , aproximadamente. La fóvea tiene casi exclusivamente conos, alrededor de 110.000.

Por otra parte, los conos de tipo L (sensibles a las ondas largas) y de tipo M (sensibles a las ondas medias) son mucho más abundantes que los de tipo S (sensibles a las ondas cortas), en una proporción de 14 a 1, aproximadamente.

En el interior de la fóvea se puede distinguir la *foveola*, una pequeña área de 0,4 mm de diámetro en la cual hay unos 25.000 conos y casi ningún bastón, y que abarca aproximadamente  $1,4^\circ$ . En el interior de la foveola también se puede distinguir una región aún más limitada denominada *isleta central*, en la cual los conos tienen una longitud máxima y no hay ningún bastón. Esta área tiene unos 0,050 o 0,075 mm de diámetro y abarca entre  $0,17^\circ$  y  $0,24^\circ$ .

En toda la retina hay unos 120 millones de bastones y de 6 a 8 millones de conos. A medida que nos acercamos a la fóvea disminuye el número de bastones hasta una zona que abarca unos  $10^\circ$ , en la cual la densidad se va equilibrando. Alrededor de la fóvea, se extiende la *mácula lutea*, un área amarillenta de unos 3 mm a 5 mm de diámetro y que abarca entre  $10^\circ$  y  $17^\circ$ . La zona que se extiende más allá de la mácula lutea contiene progresivamente mayor número de bastones y menor número de conos, como ya he dicho.

Los bastones y los conos constituyen la primera capa de células que intervienen en la percepción de la luz, la capa primaria, con mayor número de células, aunque no es la primera a la cual llega la luz, pues está situada en el nivel más profundo de la retina. Antes de llegar a esta capa, la luz atraviesa otras capas. En uno de los principales estudios de referencia, el publicado por S. L. Polyak en

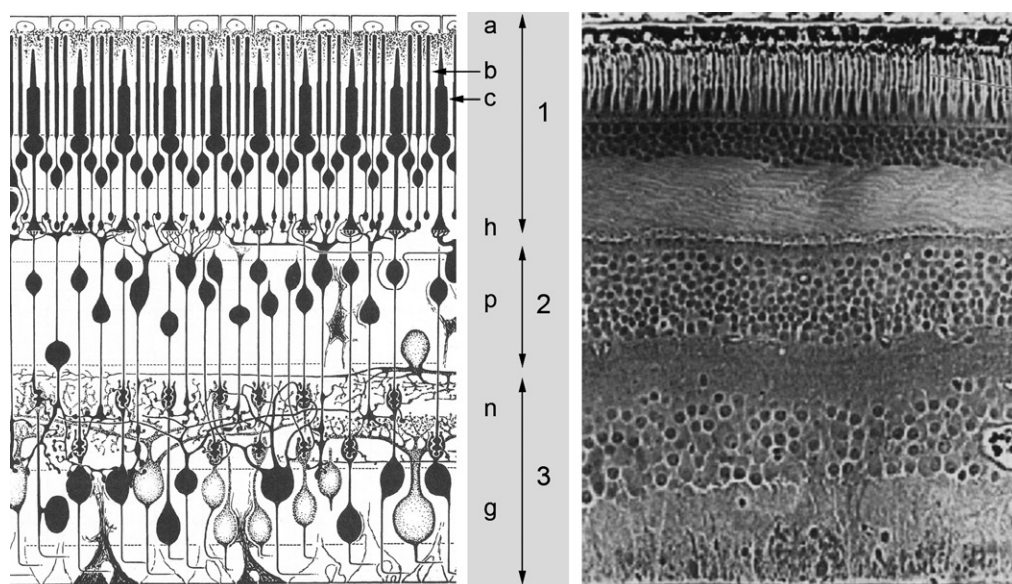


Figura 1.29 Sección de la retina mostrando las capas principales: 1 Capa de fotorreceptores: (a) epitelio pigmentado (fondo de la retina), (b) bastones, (c) conos. 2 Capa de células intermedias: (h) horizontales, (p) bipolares, (n) anacrinas. 3 Capa de células ganglionares (g). A la izquierda, un esquema adaptado de Polyak (1941). A la derecha, una imagen que muestra una sección de la retina, adaptada de Boycott y Dowling (1969) que se corresponde aproximadamente con el esquema de la izquierda. Obsérvese que el recorrido de la luz, en ambos casos, sería de abajo arriba, atravesando las capas celulares antes de activar los fotorreceptores de la retina.



1941, se distinguían hasta 10 capas diferentes que, en estudios posteriores, se agrupan en tres principales: la capa de fotorreceptores primarios (conos y bastones), la capa de neuronas intermedias (células bipolares y anacrinas principalmente) y la capa de células ganglionares. Esta estructura se sintetiza en la figura 1.29.

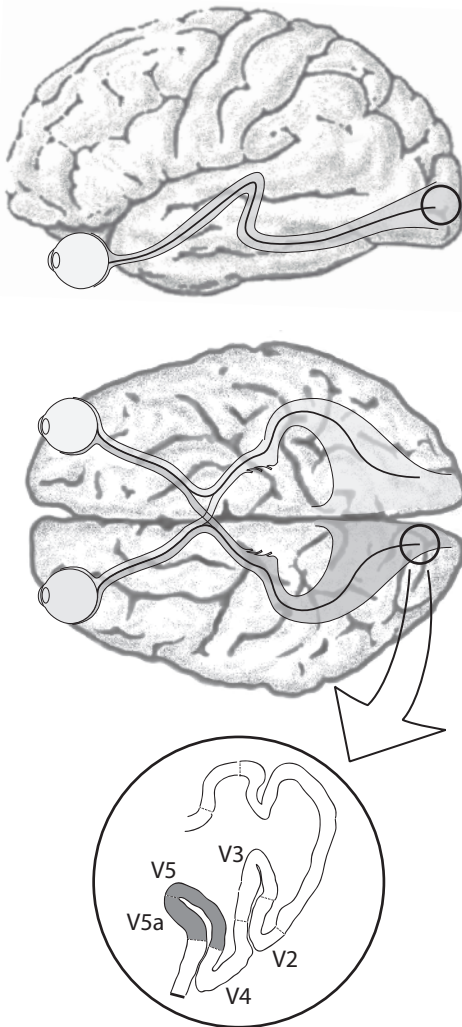


Figura 1.30 Córtex visual. Secciones verticales y horizontales del cerebro mostrando la conexión de los ojos con el córtex visual. En la parte inferior, detalle de las áreas principales.

Gran parte del procesamiento de señales lumínicas que, de algún modo, se convierten en imágenes familiares, tiene lugar en la propia retina, aunque su funcionamiento no es aún bien conocido. Pero lo que es indudable es que los impulsos que se transmiten al cerebro a través del nervio óptico ya han experimentado algún tipo de procesamiento, pues, aunque el número de células receptoras es de unos 120 millones, estas células se conectan entre sí y el número de fibras nerviosas que se canalizan a través del nervio óptico está entre 800.000 y 1 millón.

Esta canalización se hace a través del punto ciego, una zona en la que no hay receptores. Si se dibujan dos cruces en un papel y, con un ojo cerrado, mantenemos la vista fija en una de ellas mientras alejamos la vista, habrá un momento en que dejará de percibirse la otra, pues caerá en la zona correspondiente al punto ciego. El punto ciego aparece como una zona blanca en la retina debido a que, a partir de ese punto las fibras se recubren de mielina, la sustancia que recubre corrientemente las fibras nerviosas y que facilita la transmisión de impulsos.

### Mecanismos neuronales de inhibición lateral

En los análisis de la respuesta de las neuronas tiene un papel fundamental el campo de recepción asociado, es decir, el área de la retina que, al ser estimulada, provoca una respuesta de la neurona.

Muchas células ganglionares tienen una estructura característica de centro-periferia. El análisis de la respuesta de estas células se ha llevado a cabo por medio de microelectrodos insertados en la propia retina o en el nervio óptico.

Cuando el estímulo abarca todo el campo de recepción, la mayoría de las células responden con una serie de impulsos más o menos aleatorios pero de intensidad media uniforme, con una frecuencia de alrededor de 50 picos por segundo. Sin embargo, cuando el estímulo está muy localizado, la respuesta se





caracteriza por una distribución como la que se muestra en la figura 1.31. Debe tenerse presente que la intensidad del estímulo afecta a la frecuencia con que se dispara la neurona, no a su intensidad o su velocidad.

Los impulsos inhibitorios parece que vienen de las células laterales, a las cuales están conectadas las ganglionares. Cuando un receptor retiniano excita una neurona, esta manda un impulso a través del nervio óptico. Pero también envía impulsos a sus vecinas y, a través de la experiencia y del desarrollo, se forman sinapsis de conexión con estas células vecinas. Las células horizontales y anacrinas proporcionan conexiones laterales que hacen que la respuesta correspondiente a un área de la retina sea modificada debido a la respuesta de una región vecina. De este modo, un aumento de intensidad provoca, en lugar de un aumento, una disminución de la señal transmitida. Este mecanismo, por el que una parte de la retina puede actuar como inhibidor de la señal de otra parte, es lo que se conoce como inhibición lateral.

Estos mecanismos de inhibición permiten explicar el fenómeno de la constancia de la luminosidad, que es el correlato del fenómeno de la constancia cromática, otro fenómeno importante que se describe con más detalle en el libro sobre simulación de materiales. Si aumenta la iluminación recibida, las señales tenderán a intensificarse. Pero si nuestro sistema visual cuenta con un mecanismo que, al aumentar, globalmente, la intensidad de la iluminación también aumenta la intensidad de la inhibición, el resultado será que la proporción, el contraste, que es lo que nos importa para evaluar correctamente la reflectancia de la superficie de los objetos y percibirlos como algo estable, se mantiene igual. Sin embargo, en regiones más limitadas, como ocurre en nuestra percepción de los bordes de los objetos que puede ser necesario distinguir con mayor precisión, las variaciones pueden provocar un aumento de la intensidad de la respuesta sin provocar necesariamente un aumento de la inhibición, lo que nos permite mantener nuestra capacidad de discrimina-

ción de los detalles y de cómo se ven afectados por las variaciones de iluminación.

Dicho de otro modo, en palabras de David Hubel, en su discurso de aceptación del premio Nobel de 1981: "...Kuffler había descrito dos tipos de células ganglionares retinianas que denominó *on-center* y *off-center*. El campo de recepción (*receptive field*) de cada tipo consistía en dos regiones mutuamente antagónicas. En 1957, Barlow, Fitz Hugh y Kuffler habían demostrado que, como consecuencia, las células ganglionares son menos sensibles

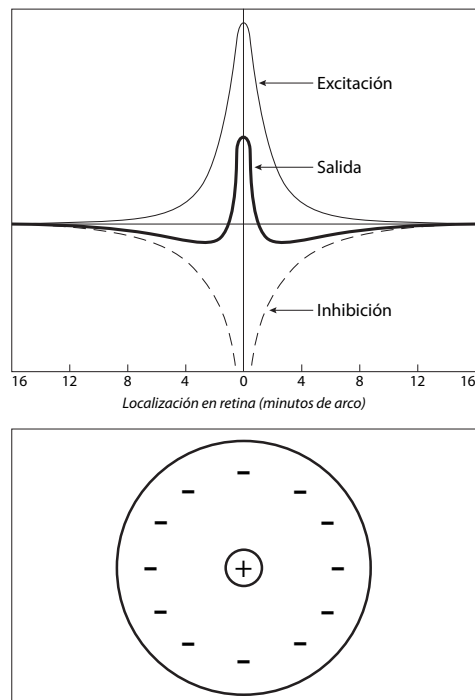


Figura 1.31 Campos receptivos característicos de células ganglionares. La figura superior muestra el efecto de un punto de luz sobre el sistema visual, calculado a partir de funciones de dispersión junto con medidas independientes de la óptica del ojo. La curva superior muestra la distribución de la excitación, y, la curva inferior, la distribución de la inhibición cuya intensidad es inversamente proporcional a la distancia. La curva central, más gruesa, muestra la distribución de salida resultante. La figura inferior muestra el campo de recepción de una fibra neuronal de centro periferia que correspondería aproximadamente a la curva de salida de la figura superior (adaptado de Cornsweet, 1970) y otros autores.

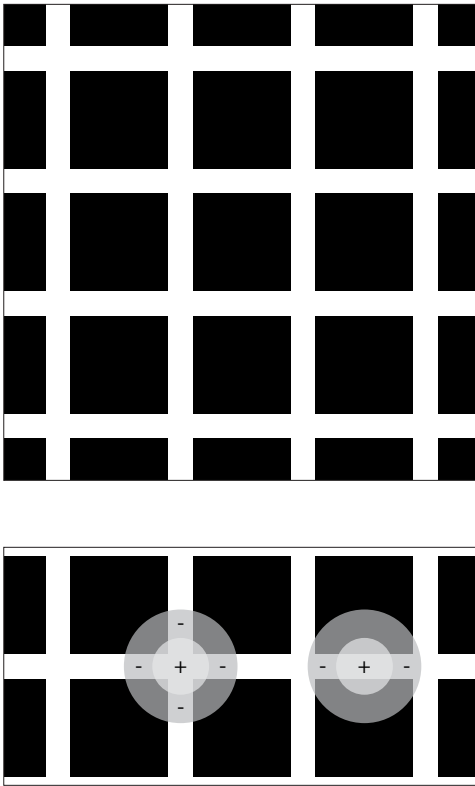


Figura 1.32 Ejemplo de ilusiones perceptivas provocadas por diferentes grados de inhibición en una red de Hermann.

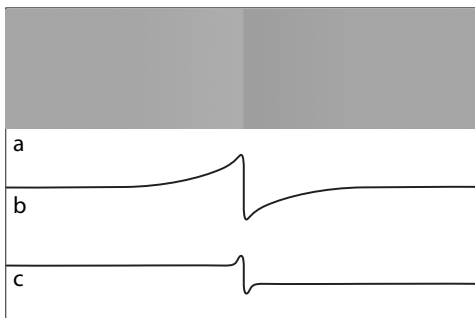


Figura 1.33 El efecto Craik-O'Brien: a) rectángulos cuya intensidad se percibe globalmente como similar, b) intensidades reales, c) intensidades percibidas.

a la luz difusa que a un punto que cubra el centro de su campo de recepción. Me llevó cierto tiempo el darme cuenta de lo que esto significaba: que el modo en que una célula responde a una escena visual cambiará poco cuando, por ejemplo, el Sol se oculte tras una nube y la luz reflejada por objetos negros y blancos disminuya por un amplio factor. La célula ignora este cambio y nuestra evaluación visual de los objetos como negros o blancos no se ve apenas afectada. Los campos receptivos centro-periferia de Kuffler comenzaron, así, a explicar por qué la apariencia de los objetos depende tan poco de la intensidad de la fuente de luz. Algunos años después, Edwin Land mostró que la apariencia de una escena es, de un modo similar, relativamente independiente de la exacta composición de color de la fuente de luz. La base fisiológica de esta independencia del color todavía tiene que ser elaborada”.

La inhibición lateral se manifiesta claramente en algunos fenómenos como las redes de Hermann (llamadas así por el fisiólogo alemán Ludimar Hermann, que informó sobre ellas por primera vez hacia 1870). En los cruces (ver la figura 1.32), la inhibición es mayor, lo que provoca una respuesta global menor, que se traduce en una percepción de menor intensidad que en las líneas intermedias, en donde la inhibición es la mitad que en los cruces.

Otro ejemplo interesante, bien conocido intuitivamente por los pintores, es lo que se conoce técnicamente como el efecto Craik-O'Brien y que se ilustra en la figura 1.33. La imagen superior (a) muestra dos rectángulos cuya intensidad se percibe globalmente como similar, aunque los bordes centrales están claramente diferenciados. Sin embargo, la mitad izquierda y la mitad derecha de ambos rectángulos tienen exactamente la misma intensidad y, a partir de esta mitad, hay un suave degradado como se aprecia en la curva que muestra las intensidades reales (b). Pero las intensidades percibidas corresponderían más a la curva que se muestra en (c). Si se cubre la línea central de los rectángulos superiores



con un lápiz, la igualdad de los extremos será claramente perceptible.

La inhibición podría entenderse como un mecanismo de abstracción. Abstraer significa, literalmente, “sacar fuera”, suprimir. Al igual que no podríamos recordar nada ni no fuéramos capaces de olvidar lo que no es relevante, tampoco seríamos capaces de formar conceptos si no fuéramos capaces de separar los rasgos primarios de los secundarios. La memoria y el razonamiento dependen de la inhibición. Y parece que el mecanismo de percepción, en su nivel más básico, recurre a mecanismos similares a los que utilizamos en los niveles superiores de nuestros procesos mentales e imaginativos.

### Receptores cromáticos profundos. El córtex visual

El córtex es una capa de unos 2 mm de espesor, con un área superficial de alrededor de 1.400 cm<sup>2</sup>, replegada en varias capas internas. Externamente, como todo el cerebro, se caracteriza por una serie de protuberancias denominadas giros o circunvoluciones cerebrales (del griego *gyros*, anillo, círculo), separadas por hendiduras denominadas surcos o cisuras. El cerebro humano incluye más surcos que el de otros primates. Los surcos más visibles se utilizan para distinguir cuatro lóbulos principales en el cerebro: el frontal (correspondiente a la frente), el parietal (la parte superior), los temporales (correspondientes a las sienes) y el occipital (correspondiente a la nuca), en donde está situado el córtex visual. La totalidad del área cortical ocupa  $1,3 \times 10^5$  mm<sup>2</sup>, con un espesor de 4 mm. Hay un total de  $10^{10}$  neuronas, aproximadamente.

A lo largo del siglo xx se ha ido comprendiendo cada vez mejor hasta qué punto la organización del cerebro está basada en áreas funcionales especializadas, con misiones tan específicas que la lesión en alguna de estas áreas equivale a la amputación de un miembro o a la pérdida de la capacidad de llevar a cabo algunas acciones elementales. Un gran número de los trabajos de investigación que

se han llevado a cabo para localizar con precisión estas áreas han tenido que ver con la visión.

Las fibras del nervio óptico que transmiten la información preprocesada en la retina, el tracto óptico, se subdividen de tal modo que la zona lateral externa de cada ojo va a parar al hemisferio cerebral del mismo lado, al denominado *campo visual ipsilateral* pero la zona lateral interna va a parar al hemisferio opuesto, al denominado *campo visual contralateral*. Esto implica un cruce que se produce en el quiasma óptico.

Más allá del quiasma, se encuentra una primera zona fundamental, una subdivisión del tálamo subcortical denominada *núcleo geniculado lateral* (NLG, por sus siglas en inglés). Este núcleo está formado por seis capas que se relacionan de modos específicos con las zonas de la retina mencionadas. El campo visual ipsilateral se conecta con las capas 5, 3 y 2, y el contralateral, con las capas 6, 4 y 1. Y puntos adyacentes de la retina, como se ha comprobado en diversos experimentos, se corresponden con puntos adyacentes en cada capa del NLG. Por otro lado, y sin entrar en la descripción de la compleja estructura de este núcleo, también se puede mencionar que las cuatro capas superiores están formadas por pequeños cuerpos celulares, por lo que se les denomina *capas P* (parvocelulares) y las dos inferiores por grandes cuerpos celulares, por lo que se les denomina *capas M* (magnocelulares). El papel del NLG, que actúa como una especie de “filtro” de la información que llega de las áreas retinianas, no se comprende demasiado bien. Se comprende mejor, sobre todo desde los descubrimientos de los últimos años, el papel de áreas más profundas.

Más allá del NLG se encuentra la corteza visual primaria o córtex visual, denominado así porque todo el procesamiento principal de las señales visuales se lleva a cabo en esta zona. El descubrimiento de estas áreas se debe, como decía al comienzo, al paciente trabajo de cientos de investigadores, entre los cuales cabe citar, en primer lugar, a David Hubel y Torsten Wiesel, premios Nobel en 1981



por sus experimentos iniciados a finales de la década de 1950.

El resultado de estos trabajos se puede resumir muy brevemente como sigue.

En primer lugar, hay una relación directa entre zonas de la retina y zonas de la corteza visual primaria, de tal modo que estas zonas reproducen un mapa de la retina en una zona determinada denominada V1 o *zona visual primaria*. La zona V1 es el área primaria del córtex, que ocupa del orden de unos 24 cm<sup>2</sup> (alrededor de 3 x 8 cm) y comprende alrededor de  $1,5 \times 10^8$  neuronas, bastantes más que las  $10^6$  del LGN. Se han identificado y analizado alrededor de 20 áreas más.

El área V1 está compuesta de seis capas principales, que se distinguen por diferencias en la densidad de neuronas, axones y sinapsis. Las tres primeras incluyen más axones y dendritas que neuronas. La capa 4 ha sido analizada más intensamente, principalmente sus conexiones con las capas parvocelulares y magnocelulares del LGN.

Las señales que llegan al córtex desde la retina mantienen ciertas características: las interconexiones se corresponden con el ojo (izquierdo o derecho) que envía las señales; se corresponden también con el tipo de célula ganglionar que envía las señales y con la posición de esta célula en la retina. Con todo, muchas neuronas responden a estímulos diversos que vienen de ambos ojos y de diferentes células ganglionares, con lo que hay un grado importante de integración que va más allá de esta correspondencia básica.

Las células ganglionares relacionadas con la fovea ocupan un área muy superior, quizás de 3 a 6 veces el área correspondiente a células periféricas. Al menos la mitad del área de la V1 representa los 10° centrales (un 2 %) del campo visual. Estas diferencias son análogas a la que se dan en otras zonas del cerebro en donde, desde hace mucho tiempo, es sabido que el área correspondiente a las manos o la lengua es muy superior al área correspondiente a otras áreas de menor importancia en el comportamiento humano.

La zona central, en torno a los 10° (aproximadamente la extensión de la mácula) está representada por al menos un 60 % del córtex visual.

En cualquier caso, el área V1 es la que contiene el mapa más detallado de la retina junto con una arquitectura funcional muy rica (neuronas especializadas en captar determinados rasgos tales como rectángulos, bordes, esquinas con determinadas orientaciones, etc.) que Hubel y Wiesel fueron los primeros en desvelar con precisión.

Por otro lado, la zona V1 está estrechamente relacionada con otras cuatro áreas denominadas, respectivamente, V2, V3, V4 y V5. La figura 1.30 muestra la ubicación de estas áreas en el córtex visual. En los últimos veinte años se ha encontrado amplia evidencia de que la información que llega de la retina se representa de manera independiente y diferente en estas áreas. La visión es, por tanto, un proceso extraordinariamente complejo en donde la información puntual, supone tan solo una parte. Estas áreas presentan cierto grado de especialización. Concretamente, la V2, aunque comparte muchas funciones de reconocimiento de patrones básicos con la V1, incorpora otras más complejas, tales como la orientación de contornos virtuales o la separación entre figura y fondo y, probablemente, el reconocimiento de patrones complejos. La V3, cuya extensión exacta aún es objeto de discusión, parece jugar un papel importante en la percepción de la forma dinámica. La V4 tiene un papel fundamental (pero no único) en la percepción del color. A partir de los trabajos de Zeki a finales de 1970, se consideró que la función principal del área V4 era la percepción del color, pero trabajos posteriores han mostrado que en ella también están localizadas funciones de reconocimiento de patrones de complejidad intermedia. Por último, la V5 tiene un papel tan fundamental en la percepción del movimiento que lesiones en esta área pueden ocasionar sorprendentes deficiencias tales como la ceguera al movimiento (los que padecen esta lesión ven



cómo los objetos cambian de posición pero son incapaces de seguir su pista).

En la zona V1, la más compleja, se han encontrado columnas de neuronas con un alto grado de especialización. Concretamente, a la orientación de pequeños segmentos, a la variación en las frecuencias espaciales, a la variación en las frecuencias temporales, a la discriminación sobre si una señal proviene de uno o de los dos ojos. Algunos de los fenómenos de percepción más complejos solo se explican adecuadamente si se tiene en cuenta esta notable especialización de los receptores situados en el córtex. La estimulación directa de determinadas zonas del córtex visual causa sensaciones de visión. Estas sensaciones de visión, que no pasan por el ojo pero que son percibidas de modo similar, se denominan *fosfenos*.

A diferencia de lo que ocurre con las neuronas del LGN que tienen, en general, campos de recepción circulares, la mayoría de las neuronas del área V1 presentan un alto grado de especialización, por lo general en una determinada dirección. Los numerosos trabajos de David Hubel y Torsten Wiesel han demostrado que hay un gran número de grupos de neuronas que se disparan al captar orientaciones específicas, horizontales, verticales, principalmente, pero también con diversos ángulos. Estos grupos de neuronas incorporan campos de excitación que se agrupan linealmente y que discurren en paralelo a campos laterales de inhibición. Hubel y Wiesel también encontraron grupos especializados en otros aspectos tales como la dirección del movimiento. Todo esto apunta a un grado de especialización que choca con la idea de que la percepción es un acto unitario, global. Y no todos los investigadores han aceptado este tipo de conclusiones.

### Una visión global del cerebro

El descubrimiento de que en el córtex visual hay grupos de neuronas especializadas en captar determinados rasgos del entorno ha supuesto, sin duda, un espectacular avan-

ce en la comprensión del funcionamiento de nuestro sistema de percepción del mundo. Sin embargo, hay muchos fenómenos que quedan sin explicación. Entre los más notables está el hecho desconcertante de que, si un ser humano sufre una lesión en estas zonas, pierde toda sus capacidades funcionales, es incapaz de orientarse adecuadamente. Si a un gato o a un mono les ocurre lo mismo, en los primeros días su comportamiento es igual al del hombre, se convierten en animales inválidos. Sin embargo, con el tiempo recuperan buena parte de sus capacidades y los monos que han sufrido, experimentalmente, la extirpación de alguna de esas zonas, acaban por correr y subirse a los árboles prácticamente igual que lo hacían antes. De hecho, también ocurre hasta cierto grado lo mismo con el hombre. Sujetos que han sufrido accidentes traumáticos en el cerebro han acabado, en muchos casos, por recuperar la funcionalidad perdida. Todo ocurre como si el cerebro almacenase, de modo todavía desconocido, información adicional que permite recuperar la capacidad de percepción más allá de estas áreas especializadas.

El científico Karl Pribram formuló una atrevida hipótesis (véase Pribram, 1971) según la cual, muchos de los fenómenos del funcionamiento cerebral podían explicarse si se considera que el funcionamiento del cerebro es más parecido a un holograma que a un mecanismo de elementos discretos conectados entre sí por miríadas de cables. Otro gran científico, David Bohm, ha formulado otras hipótesis no menos atrevidas dirigidas, en su caso, a una explicación global del universo. Bohm considera que hay un “orden implicado” en el universo que presentaría notables analogías con el principio holográfico. En esta concepción, las partículas y los elementos que percibimos como discretos (como un “orden explicado”), surgirían a partir de patrones de interferencia de ondas de energía. Pribram y Bohm colaboraron en un modelo honómico del funcionamiento cerebral.



Los primeros hologramas fueron creados por el físico húngaro Dennis Gabor en 1947 (que recibió el premio Nobel de Física por este descubrimiento en 1971), pero no pasaron de un descubrimiento de interés más bien teórico hasta la aparición del láser, lo que permitió a otros dos científicos, Emmett Leith y Juris Upatnieks, en 1963, crear hologramas por medio de una serie de recursos que, pocos años más tarde, permitirían a cualquier aficionado provisto de un equipo no demasiado complicado replicar este proceso para obtener imágenes “tridimensionales”, imágenes que mostraban diferentes partes del objeto original al desplazar el punto de vista del observador.

Los hologramas se crean iluminando una placa con dos rayos de luz coherente (preferentemente de un láser), uno dirigido directamente a la placa y otro, al objeto, y redirigido hacia la placa por medio de un espejo u otro dispositivo adecuado. Las ondas de luz de los dos rayos crean un patrón de interferencia que es lo que se graba en la placa. Una de las notables peculiaridades de los hologramas es que todos los puntos de la placa (que no muestran ningún patrón reconocible, a diferencia de lo que ocurre con una placa fotográfica convencional) guardan información sobre el objeto de referencia, información transmitida por las ondas de luz reflectada que provienen del objeto desde múltiples direcciones. El holograma, por tanto, como indica su propio nombre, supone una grabación de la totalidad de la escena de referencia. Pero, por añadidura, la capacidad de almacenamiento de información está tan solo limitada por la longitud de onda, y esta capacidad es de varios órdenes de magnitud superior al de cualquier sistema convencional de grabación de imágenes.

Pribram especuló con la posibilidad de que el principio holográfico, que no está ligado necesariamente al procedimiento utilizado corrientemente para generar hologramas, pudiera explicar el modo en que la información se transmite en el cerebro por

medio de “ondas cerebrales” (*brain waves*), que permitirían la transmisión de información entre neuronas, por medio de patrones de interferencia que provendrían de, al menos, dos fuentes diferentes, y de un modo que pudiera explicar toda una multitud de fenómenos que no quedaban, y no quedan, suficientemente explicados por las teorías recientes.

Baste esta breve nota, antes de proseguir, para salir al paso de una visión excesivamente mecanicista del cerebro y los procesos de percepción. Si hay muchas cosas nuevas que se han descubierto hay muchas otras que quedan por descubrir, y hay múltiples aspectos de los fenómenos perceptivos que no encajan en las descripciones de detalle y que no deberían perderse de vista.

## 1.7 La formación de imágenes

### Límites geométricos. Capacidad inicial de resolución del sistema visual

La determinación del poder de resolución o agudeza visual del ojo es un tema importante para el que tampoco cabe una respuesta unívocamente determinada. Esto es debido a que es preciso ponderar al menos los siguientes factores: a) las condiciones de adaptación de la retina a las condiciones de iluminación; b) la intensidad de iluminación del objeto observado; c) el contraste o relación entre la luminancia del fondo y la luminancia del objeto observado; d) el lugar de la retina en donde se efectúe la medición; e) la proyección subjetiva de expectativas de reconocimiento de una figura determinada; f) la propia capacidad o poder de resolución del sujeto, que varía considerablemente de unos individuos a otros y con la edad.

Además de todo esto, es preciso establecer un método de medición y unas unidades de medida. La literatura científica sobre el tema proviene de varios campos y, como sería de esperar, es muy abundante. Dado que



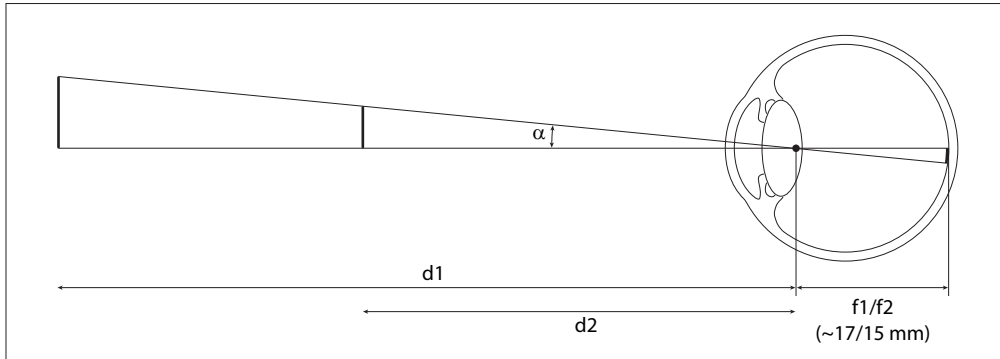


Figura 1.34 Formación de imágenes en el ojo a corta y larga distancia.

lo que aquí interesa es acordar unos valores de referencia, limitaré al máximo el análisis de la problemática especializada para resumir lo más brevemente posible el modo en que estos valores afectan a la medición del poder de resolución en los humanos y en las máquinas. Pero todos los aspectos implicados tienen mucho interés, lo que justifica que le dediquemos algo de espacio.

Se puede abordar el tema desde cuatro puntos de vista principales: mediante razonamientos puramente geométricos, mediante datos que provienen de la psicología experimental, mediante datos que provienen de una especialidad relevante para este caso, como es la oftalmología, y desde la experiencia de los fotógrafos. Veremos que los resultados son bastante similares.

1. Desde un punto de vista puramente geométrico, podemos razonar como sigue. Para que dos objetos pequeños muy cercanos puedan percibirse como distintos, parece lógico pensar que deben estimular al menos dos receptores retinianos diferentes. Sabemos que la capacidad de discriminación es mayor en la región central de la fovea, en donde solo hay conos, y que en esta área el tamaño de los conos, que están yuxtapuestos, es del orden de 2 micras.

La formación de las imágenes en el ojo depende de la distancia focal del ojo. Esta distancia varía con la acomodación del cristalino a objetos distantes o cercanos, y varía con la edad y las diferencias individuales, y

varía también según las diferentes fuentes. Pero para una descripción general, estas diferencias no son relevantes. Simplificando los datos citados, podemos aceptar que hay un punto nodal o focal principal que estaría situado en la posición marcada en la figura 1,34, dentro del cristalino y del lado de la retina. Para un objeto situado en un punto remoto, la distancia focal, desde el punto nodal a la fovea, sería de unos 17 mm (16,68 en los datos de LeGrand). Para un objeto situado en un punto cercano, a la distancia considerada como distancia normalizada de observación cercana, 25 mm, la distancia focal sería de unos 15 mm (14,77 en los datos citados).

Un cono foveal de unas 2 micras (0,002 mm) abarcará entre  $0,45'$  y  $0,40'$  pues

$$\text{arc tg de } 0,002/15 = 0,000133\text{rad} = 0,0076^\circ = 0,45'$$

$$\text{arc tg de } 0,002/17 = 0,000117\text{rad} = 0,0067^\circ = 0,40'$$

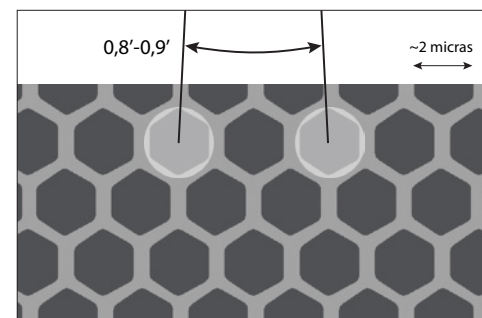


Figura 1.35 Ángulo cubierto por dos conos foveales.



Esto quiere decir que un objeto luminoso que subtendiera este ángulo podría llegar a ser percibido en circunstancias óptimas. Pero lo que nos interesa es la capacidad de resolver dos puntos muy próximos. Si están tan próximos que sus rayos, prolongados de modo que pasen por el centro focal del ojo, caen sobre un mismo cono serán percibidos como un único punto. Parece que será necesario que caigan al menos sobre dos conos contiguos para ser percibidos como puntos distintos.

El límite de discriminación, en consecuencia, deberá estar relacionado con la distancia angular entre dos conos contiguos. Para una distancia focal de 14 a 17 mm y conos de 2 micras (para relacionarlo con valores anteriores), la distancia será de 4 micras, por consiguiente:

$$\text{arc tg de } 0,004/15 = 0,000266 \text{ rad} = 0,0152^\circ = 0,91'$$

$$\text{arc tg de } 0,004/17 = 0,000235 \text{ rad} = 0,0134^\circ = 0,81'$$

Es decir, algo menos de 1' de grado. Para hacernos una idea de lo que esto quiere decir, podemos extender el brazo todo lo que podamos y mirar la uña del dedo pulgar. Esto abarca aproximadamente un grado.

Sin embargo, estos cálculos geométricos deben ponderarse, como veremos, por los límites impuestos por la difracción y por otros factores que contribuyen a dispersar la formación puramente geométrica.

Antes de continuar no estará de más comparar estos datos con los que provienen de la observación de objetos naturales como el Sol

y la Luna y las estrellas, que son referencias universales que nos ayudarán a situar estos datos.

El ángulo subtendido por el Sol se puede calcular con facilidad del mismo modo que en el caso anterior. El diámetro del Sol es de 1.392.000 km. La distancia de la Tierra al Sol en el perihelio (punto más cercano de la órbita de la Tierra al Sol, a principios de enero) es de 147,5 millones de kilómetros. En el afelio (punto opuesto al perihelio, a principios de julio) es de 152,6 millones de kilómetros. Si tomamos como distancia media del Sol a la Tierra 149,6 millones de kilómetros, de estos valores resulta que el diámetro angular en el perihelio es de  $32'35,64''$  y en el afelio de  $31'31,34''$ . Redondeando, se puede considerar un diámetro angular medio de  $32'$  (o  $0,53^\circ$  o  $0,0093$  radianes). Otra forma de decir lo mismo es que vemos el Sol con un ángulo de  $1/108$  radianes. Lo vemos igual, por tanto, que si a 108 cm de distancia tuviera una anchura de 1 cm, o a 108 km, una anchura de 1 km.

Y otro tanto con la Luna. El diámetro de la Luna es de 3.476 km. La distancia de la Tierra a la Luna en el perigeo (más cerca) es de 356.425 km, y en el apogeo (más lejos) de 406.710. De estos valores resulta que el diámetro angular en el perigeo es de  $33'28''$  y, en el apogeo,  $29'23''$ . Si tomamos como distancia orbital media 384.000 km resulta un diámetro angular medio de  $31'52''$  o  $32'$ . Prácticamente el mismo que el del Sol, algo que no debería sorprendernos si recordamos lo que ocurre con los eclipses.

Como es bien sabido, la Luna parece más grande cuando está cerca del horizonte. Las mediciones demuestran que no cambia de tamaño: el ángulo cubierto es el mismo cuando está cerca del cenit que cuando está cerca del horizonte. Parece claro, por tanto, que es un fenómeno psicológico, debido principalmente a que cuando el Sol y la Luna están en el cenit no hay objetos con los que compararlos y parecen muy lejanos, algo que no ocurre cuando se muestran, debido a su luminosidad, con el mismo aspecto que objetos familiares pero

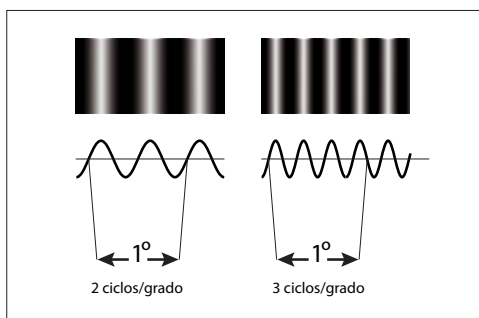


Figura 1.36 Ciclos por grado.





que vemos muy empuqueñecidos por la distancia. Esto también puede ser medido y las mediciones muestran que los sujetos estiman la diferencia de tamaño en cifras del orden de 2,5 a 3,5 veces mayor (al anochecer y con cielo semicubierto el efecto es aún mayor).

Pero lo más interesante en relación con esta discusión es el tamaño de las estrellas, pues las tenemos de todo tipo y podemos calibrar nuestra capacidad de discriminación en relación con su tamaño. Sin embargo, aquí surge otro de los factores que mencionaba al principio. En el límite, la capacidad de resolución está estrechamente relacionada con la capacidad de discriminación, por lo que aplazaré esta discusión para el apartado siguiente.

2. La *psicología experimental* proporciona varios datos que se pueden comparar con los anteriores. Podemos incluir en este apartado los datos proporcionados por los astrónomos y que se remontan a muchos siglos atrás. La cifra de 1 min de grado, que ya aparece en las observaciones de astrónomos de la antigüedad y que es, aproximadamente, la misma que dió Helmholtz en el siglo pasado, se toma corrientemente como límite normal de la agudeza visual.

Según otros datos aportados por la psicología experimental, dos líneas de 1 mm comienzan a fundirse en una mancha continua a partir de los 4 metros por término medio, pero en condiciones óptimas pueden llegar hasta los 7 metros. Este dato depende de la capacidad del sujeto, y es variable. Es un dato muy similar al que hemos obtenido más arriba ( $2 \text{ mm}/7.000 \text{ mm} = 0,000285 \text{ rad} = 0,016^\circ$ ), algo menos de  $1'$ , por lo que podemos aceptar que los datos proporcionados por los experimentos coinciden con las deducciones teóricas.

Si razonamos a la inversa, aceptando  $1'$  como referencia adecuada, que equivale a  $0,01666^\circ$  o  $0,00029$  radianes, podemos preguntarnos cuál será la anchura de una trama que podemos resolver a la distancia de observación  $D_v$  (250 mm), considerada como normal para observación a corta distancia y en condiciones adecuadas, esto es, buena iluminación y un objeto presentado con contraste

suficiente. La respuesta será  $0,0725 \text{ mm}$  ( $250 \text{ mm} \times 0,00029$ ). Esto quiere decir que en 1 cm entrarían 137 líneas (y en una pulgada, 347), otro dato al que volveremos en el capítulo siguiente a la hora de hablar de la resolución adecuada de dispositivos virtuales. Pero cualquiera que haya tenido que imprimir una imagen sabe que las resoluciones gráficas recomendadas son, precisamente, de ese orden: 300 dpi (*dots per inch*), equivalentes a unos 120 ppc (puntos por centímetro).

3. En *oftalmología* se utilizan principalmente las escalas de Snellen, propuestas por Hermann Snellen (1834-1908) en 1862, o los anillos de Landolt, propuestos por Edmund Landolt (1846-1926) hacia 1890. La más utilizada es la primera, que se basa en tres criterios principales: a) la utilización de las letras del alfabeto como patrón de reconocimiento; b) la aceptación de 1 min de arco como poder "normal" de resolución; c) el establecimiento de una distancia de 20 pies (unos 6 metros) como práctica desde el punto de vista clínico. A esta distancia (que requiere  $0,17d$  de acomodación), cada elemento, tal como el palo de una "e", de una de las letras utilizadas en las cartillas normalizadas, subtiende un arco de  $1'$ , y la letra completa, un arco de  $5'$ . El numerador de la notación Snellen indica la distancia de observación (medida en pies). El denominador indica la distancia a la cual un elemento de la letra que ha sido reconocida subtiende un ángulo de  $1'$  de arco. El numerador suele ser habitualmente 20 (pies), pero puede ser cualquier otra distancia. Así, el valor 20/20 indica un ángulo de  $1'$ . El valor 20/10, el más bajo que se encuentra en esta notación, equivale a la mitad del anterior,  $0,5'$ . El valor 20/40 a  $2'$ . Y así con los demás: cualquier otro valor puede obtenerse por la sencilla fórmula  $a = 1/S$  en donde "a" será el ángulo en minutos, y S, el valor del cociente Snellen.

Estos datos coinciden también con los anteriores. Pero las experiencias oftalmológicas indican que muchos sujetos jóvenes con buena visión pueden discriminar adecuadamente las letras de prueba con un án-



gulo inferior, de 0,5 minutos de arco. Y este valor se acepta actualmente como límite de la agudeza visual.

4. En *fotografía* y otros contextos similares se utiliza la noción de enrejados (*gratings*) o bien, más recientemente, una herramienta matemática denominada MTF (*Modulation Transfer Function*). Un enrejado es una trama de franjas claras y oscuras. En estos casos, se acostumbra a dar los valores en ciclos por grado de ángulo visual para un determinado nivel de contraste. Un ciclo viene dado por una franja oscura y una clara, con lo que tendremos 60 ciclos por grado como medición equivalente a 1 minuto de arco.

Un modo sencillo de apreciar estos valores, siguiendo con el ejemplo dado anteriormente, es extender el brazo y mirar la anchura de la uña del dedo meñique. Esto es aproximadamente lo que abarca un grado (1 cm a 57 cm de distancia). Si pintáramos la uña con una franja vertical negra y otra blanca (o sostuviéramos un papelito así pintado a esa distancia), tendríamos un ciclo por grado que es la unidad de referencia que se utiliza en *gratings* de este tipo. Y si pintáramos 4 franjas, 2 blancas y 2 negras, tendríamos 2 ciclos por grado y así sucesivamente hasta llegar a los 60.



Figura 1.37 Discos de Airy.

La razón por la cual se utiliza este sistema es porque incorpora un aspecto fundamental que hasta ahora hemos dejado a un lado: el contraste. Hasta aquí hemos hablado de la resolución o la agudeza visual como la capacidad de distinguir “líneas negras y blancas” y “en condiciones óptimas”. Pero esto es muy vago, ¿qué pasa si las condiciones no son óptimas o las líneas no son negras y blancas sino oscuras y claras? La ventaja de utilizar un sistema que tenga en cuenta estos factores es que se acerca bastante mejor a los muy diferentes casos que se dan en la práctica.

Pero esto implica adentrarse en otro terreno aún más complejo: el de la discriminación de valores que es de lo que tratarán los apartados siguientes.

Para concluir, podemos recordar que, provisionalmente y en condiciones similares, las notaciones “1 minuto” (arco subtendido), “20/20” (notación de Snellen) y “60 c/grado” (o cpd, notación de enrejados) son equivalentes. Estas condiciones son una buena iluminación, contraste óptimo y 2° de visión (foveola). Más allá de esta zona central la agudeza visual desciende con rapidez.

### Poder de resolución y difracción. Círculos de confusión o círculos de Airy

Antes de cerrar este apartado, cabe añadir un par de observaciones que implican pasar de la óptica geométrica a la óptica física. Hay tres factores que modifican los resultados obtenidos a partir de razonamientos exclusivamente geométricos: la difracción, las aberraciones ópticas y la dispersión.

En primer lugar, si afinamos un poco más los cálculos anteriores, resulta que cada sistema óptico, debido a la difracción, tiene una capacidad limitada de formar imágenes detalladas de un objeto. La difracción pone límites al tamaño mínimo que la pupila permite que se proyecte sobre la retina. Y la precisión de los bordes depende de un equilibrio entre la difusión y la difracción. A medida que el diámetro de la pupila aumenta, la difusión lineal también aumenta. Por otro lado, a medida



que el diámetro de la pupila disminuye más allá de cierto límite, que estaría en torno a los 3 mm de diámetro, la difracción aumenta. El valor óptimo de equilibrio entre estos dos factores opuestos estaría en una pupila de 2,4 mm de diámetro.

El poder de resolución, esto es, la capacidad de percibir distintamente dos pequeños círculos o líneas yuxtapuestas, se puede medir por medio de la fórmula de Lord Rayleigh:

$$\gamma = 1,22 \lambda / D$$

donde  $D$  es el diámetro de la abertura considerada,  $\lambda$ , la longitud de onda de la luz,  $\gamma$  la magnitud angular del círculo resultante. A partir de aquí, podríamos calcular el tamaño del círculo resultante.

Sin embargo, esto se complica debido al efecto de difracción. La imagen proyectada por un punto de luz que atraviesa una lente puntual no forma un único círculo sino una serie de anillos concéntricos. Esto es lo que se denomina un *Airy disk* o un *Airy pattern*. Ambas denominaciones vienen del matemático y astrónomo inglés Georges Biddell Airy (1801-1892) que fue, entre otras cosas, el primero en describir este fenómeno hacia 1835. A una distancia suficientemente grande de la apertura, el ángulo en que se da el primer mínimo de la serie de círculos viene dado aproximadamente por la fórmula

$$\sin \theta = 1,22 \lambda / d$$

que, para ángulos pequeños, es aproximadamente igual a

$$\theta = 1,22 \lambda / d$$

en donde  $\theta$  es el ángulo en radianes,  $\lambda$ , la longitud de onda y  $d$ , el diámetro de la apertura.

Así, se puede obtener este primer valor que a veces se denomina, un tanto equívocamente, el “radio del disco de Airy”. Sin embargo, los diámetros de los siguientes círculos dependen de la intensidad de la luz. Y como los medios de registro tienen diferentes umbrales de sensibilidad, el número de círculos también puede variar.

Con todo, el valor de este primer círculo es una referencia importante para saber hasta dónde llega la capacidad de resolución efectiva del ojo o de una cámara.

En el caso de una cámara, aplicando la fórmula anterior, podemos hacer el ángulo  $\theta = x / f$ , que es válida para ángulos muy pequeños y en donde  $x$  sería el diámetro de un punto de luz (o un minúsculo objeto) y  $f$ , la distancia focal. De esto resulta que  $x = 1,22 \lambda f/d$ . Pero el valor  $f/d$  es el número  $f$  de una cámara con el que indicamos la apertura del diafragma. Si tomamos como longitud de onda mínima la correspondiente a la región de las ondas cortas, unos 420 nm, y tomamos como apertura del diafrag-

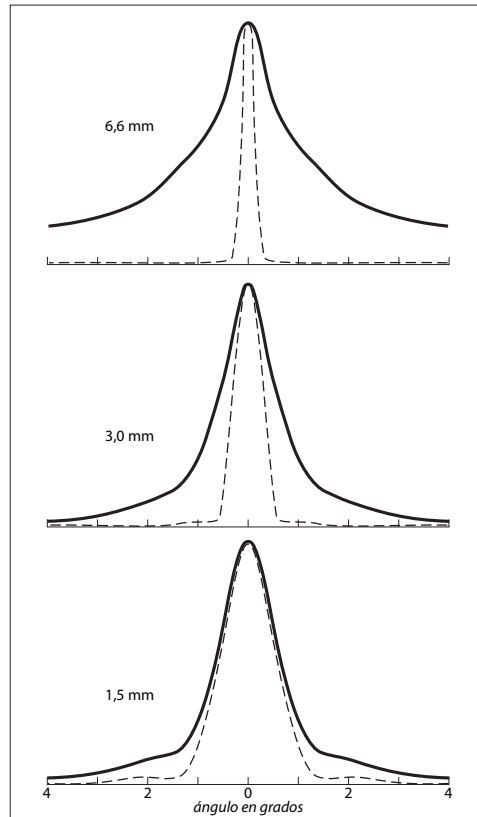


Figura 1.38 La función de dispersión de una línea (Line Spread Function) del ojo humano. La línea discontinua muestra el límite correspondiente a la difracción. Adaptado de Campbell y Gubisch (1966).



ma un valor medio corriente, como  $f/8$ , resultará que  $x = 4$  micras aproximadamente ( $1,22 \times 420 \times 8 \times 10^{-9}$ ). Esto quiere decir que si se hacen los píxeles del sensor de la cámara más pequeños de este valor no aumentará la resolución de la imagen.

Si la distancia focal fuera, por ejemplo, 50 mm, el valor anterior implicaría que la apertura es 6,25 mm (pues  $N = df/D$ , de donde  $D = df/N = 50/8$ ). En el caso del ojo humano, si, operando al revés, consideramos una distancia focal,  $df$ , de 17 mm y una apertura de pupila de unos 3 mm, tendremos que  $N = 5,6$  ( $17/3$ ). Repitiendo el cálculo anterior resultará un valor de unas 2,8 micras, ligeramente superior al de los conos foveales.

### Funciones de dispersión en el ojo

La segunda observación tiene que ver con el papel de las aberraciones y la dispersión.

Los rayos de luz que penetran en el ojo se desvían de su trayectoria debido a dos fenómenos adicionales: las aberraciones o imperfecciones del sistema visual y la dispersión.

Las aberraciones o imperfecciones de la lente hacen que los rayos no se refracten exactamente igual al atravesar el cristalino, lo que hace que los rayos que irían a parar a un único punto acaben en un pequeño círculo en torno al punto ideal.

La dispersión es debida al paso de la luz por el medio ocular que, al igual que ocurre en otros casos que ya hemos visto, hace que los rayos sufran cierto grado de dispersión.

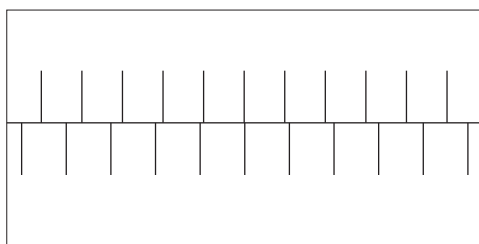


Figura 1.39 Agudeza Vernier.

Esta dispersión es tanto mayor cuanto mayor sea el diámetro de la pupila.

Las funciones de dispersión debidas a un punto y una línea son un instrumento clave en los análisis ópticos desde mediados del siglo xx. Hay dos funciones características a las que volveremos más adelante: la función de dispersión de un punto, PSF por sus siglas en inglés (*Point Spread Function*) y la función de dispersión de una línea, LSF (*Line Spread Function*).

Las primeras medidas precisas, que siguen siendo una referencia principal, se publicaron por Campbell y Gubisch (1966). La figura 1.38 muestra una breve selección de estos resultados con la intención de subrayar que la conversión de un punto en un círculo, debida a la difracción, debe corregirse tanto más cuanto mayor sea la apertura de la pupila.

### Detección y reconocimiento. Agudeza Vernier

No hay que confundir *detección* con *reconocimiento*. En las pruebas clínicas, el sujeto puede reconocer la letra “A”, sin llegar a distinguir la barra horizontal, porque entreve un triángulo invertido y sabe que no hay otra letra con este aspecto general. Pero le puede costar más trabajo distinguir otras letras. En este sentido, adquirió particular importancia, a finales del siglo xix, la denominada “C Landolt” diseñada por Landolt, a quien ya me he referido anteriormente, como un anillo interrumpido por un corte de 1 minuto de arco ( $1/5$ ) del diámetro total que subtiende, a su vez, 5 minutos y que se presenta en las cuatro direcciones para detectar anomalías específicas. Este test fue abandonado posteriormente pero sirvió de base para la definición de la agudeza visual, en 1909, como “recíproca de la cuerda angular umbral de la abertura, en minutos de arco, de un círculo interrumpido Landolt, negro, sobre fondo blanco”. En cualquier caso, el punto que quiero subrayar aquí es que determinas formas se reconocen por



familiaridad, aunque sus rasgos concretos no se detecten.

Por otro lado, aunque no haya reconocimiento, la capacidad de detección también varía según el tipo de estímulo. Concretamente, la capacidad de discriminación aumenta de un modo espectacular se trata de detectar la falta de alineación de dos segmentos lineales. Esto se denomina “hiperagudeza” y también *agudeza Vernier*. El nombre parece que deriva del matemático francés Pierre Vernier (1580-1637), que inventó la escala que lleva su nombre de la cual deriva el calibrador denominado *nonius* o “pie de rey” en español, y cuya lectura se facilita precisamente por el fenómeno de la hiperagudeza. Los valores que se han encontrado experimentalmente miden la capacidad para alinear dos segmentos lineales, algo que los seres humanos conseguimos hacer con una habilidad que va más allá de las capacidades estrictamente geométricas de nuestro sistema visual. Según diversos experimentos, la agudeza Vernier es de alrededor de 0,13 minutos de arco, lo que supone de 5 a 10 veces menos que los valores extremos que, en condiciones óptimas, no regulares, llegarían como máximo a los 0,4 minutos de arco, y en condiciones corrientes, como ya hemos visto, a 1’ de arco.

Dado que esta capacidad no puede estar basada en los fotorreceptores retinianos, pues es inferior a lo que puede ser captado por un único cono, se considera que es una capacidad que se ejerce desde el córtex visual. Esto se corresponde con los descubrimientos de ciertas zonas del córtex visual en las que se han encontrado columnas de neuronas especializadas en captar exclusivamente diferencias de orientación de bordes, y que están relacionadas con grupos de conos del mosaico retiniano que están alineados entre sí.

También se ha demostrado que esta capacidad aumenta en los meridianos horizontal y vertical, lo que también se corresponde con la especialización a determinadas orientaciones de columnas de neuronas encontradas en el córtex visual. En general, la agudeza visual

suele ser mejor en las direcciones horizontal y vertical, lo que sugiere una cierta especialización de los receptores retinianos o de los receptores neurológicos a estas orientaciones, debida, probablemente, a que la evolución ha favorecido a los sujetos con estas capacidades.

También la capacidad para detectar una línea oscura sobre un fondo claro es mayor que la capacidad que se podría esperar de la acción exclusiva de los receptores retinianos. En condiciones óptimas, esta capacidad es incluso menor que la hiperagudeza Vernier. Pero se ha comprobado que la anchura de la línea no se percibe propiamente sino que lo que se capta son minúsculas diferencias de contraste. De hecho, si se disminuye la anchura de la línea lo que se percibe es una disminución de intensidad, no una disminución de la anchura.

Además de las cuestiones relativas al reconocimiento hay otros factores a considerar. El ángulo subtendido por un cono foveal es, redondeando hacia arriba, 0,5’ de arco (o unos 30”), como ya hemos visto. Esto implica que para poder distinguir dos puntos, estos deberán caer al menos en dos conos contiguos, lo cual supone al menos 1’ de arco, que coincide con lo que ya hemos visto. Sin embargo, una línea se puede distinguir aunque el ángulo de observación sea bastante menor, un hecho intrigante al que se han buscado varias explicaciones, incluidas las puramente físicas.

El caso es que una línea que abarcase no mucho más de 1” podría ser detectada por nuestro sistema visual pues bastan unos pocos cuantos de luz, como veremos más adelante, para estimular un receptor retiniano. Pero no podría ser “vista” como tal línea. Para que fuera vista debería tener al menos alrededor de 30’ de longitud para que estimulase una serie de conos, quizás de unos 60’, que enviasen información a las células ganglionares capaces de detectar alineaciones significativas.

Hay otro factor que contribuye a esta detección. Cuando la luz alcanza la retina se dispersa. Y, como también hemos visto, en general

esta dispersión se traduce en la creación de pequeños círculos de tamaño mínimo cuya distribución se ha dado en las figuras del apartado anterior. Se ha especulado con que variaciones en la distribución dada por estas curvas podría ser alterada por variaciones inferiores a los ángulos de recepción mínimo y que estas variaciones podrían ser detectadas por el sistema visual, lo que contribuiría al proceso de detección. En cualquier caso, sea cual sea la explicación, es un hecho comprobado que en determinadas circunstancias podemos detectar líneas significativas que quedarían por debajo de los umbrales teóricos de captación dados por la óptica geométrica.

Para volver al tema del reconocimiento hay que insistir, para concluir, en que la percepción está estrechamente relacionada, por un lado, con la memoria, y, por otro lado, con la tendencia a priorizar ciertas formas por encima de otras, formas con una “buena gestalt” para usar el término propio de la teoría de este nombre, la *Gestalt Theory*, surgida en Alemania durante el primer cuarto del siglo xx y muchos de cuyos principios siguen siendo válidos.

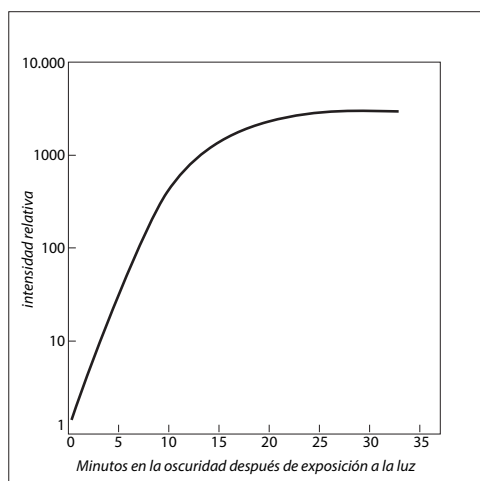


Figura 1.40 Variación de la sensibilidad visual en función del tiempo de adaptación a la oscuridad después de una exposición a una luz brillante (adaptado de Kohlrausch, 1931).

## 1.8 La formación de imágenes. Límites de discriminación de valores

La apreciación visual de los valores, de la distribución de luminancias en una escena, se relaciona también con varios temas complejos: la percepción de diferencias, la adaptación a los diferentes rangos a que nuestro sistema visual es sensible, la medición del contraste, la construcción de escalas con intervalos regulares. En esta sección, intentaré resumir del modo más completo posible los desarrollos científicos que han posibilitado contar con conceptos y recursos más eficaces para controlar las imágenes virtuales y en los que se basan algunas de las técnicas más sofisticadas que se abordarán en los capítulos siguientes.

### Condiciones de adaptación

Como decía al comienzo del apartado anterior, los datos mencionados están referidos a condiciones “óptimas”. Este es un término muy ambiguo y que requiere más explicaciones.

La curva de la figura 1.40 muestra un estudio temprano de este fenómeno que sigue siendo válido. El sujeto se expuso a una luz brillante durante unos minutos. Luego se apagó la luz y se midió su sensibilidad a destellos breves de luz de muy baja intensidad. Como se puede apreciar en la figura, la sensibilidad aumenta rápidamente durante los primeros 15 minutos y sigue aumentando más lentamente hasta estabilizarse a los 30 o 40 minutos. Corrientemente, se considera que 40 minutos de adaptación es lo recomendable para medir la sensibilidad en condiciones fotópicas, pues con esta duración la adaptación es completa.

Sin embargo, un análisis más fino muestra problemas que conviene tener presentes. Las primeras observaciones en psicología experimental, a mediados del xix, establecieron que la percepción de las diferencias lumínicas depende de los niveles de intensidad, esto es, que el intervalo de cambio de intensidad apreciado se hace progresivamente mayor a medida que aumenta la intensidad del estímulo.





mulo, lo que se conoce como ley de Fechner-Weber, que comentaremos más adelante. Por añadidura, al variar las condiciones de adaptación y el ángulo de observación, variaban también las respuestas. Esto implica dificultades casi insuperables a la hora de relacionar la capacidad de discriminación con las condiciones de observación. Este tema tiene gran importancia tanto teórica como práctica por lo que también convendrá dedicarle alguna atención.

Hurvich y Jameson describen un experimento que ilustra con cierta comicidad esta situación. Se trataría de determinar cuál es el umbral, el límite diferencial inferior, a partir del cual una fuente puntual de luz es visible sobre un fondo oscuro. Para ello, se dispone el escenario siguiente. En una habitación oscura se sitúa el sujeto del experimento, sentado inmóvil sobre una silla y mirando fijamente un agujero de unos 5 cm de diámetro. Sobre el agujero se ha dispuesto una lente de dispersión que asegure la homogeneidad del estímulo. Al otro lado, en otra habitación, se sitúa un foco diminuto, montado sobre un carril y apuntando al agujero. Como sabemos que la intensidad disminuye de modo inversamente proporcional al cuadrado de las distancias, es posible medir con precisión la disminución de la intensidad proyectada sobre el agujero sin necesidad de recurrir a otros dispositivos más sofisticados (como potenciómetros, etc.), pero que podrían alterar el tipo de luz y, por consiguiente, la linealidad de la respuesta. El sujeto se comunica con el experimentador por medio de un teléfono instalado entre las dos habitaciones.

El experimentador comienza el experimento haciendo retroceder lentamente, por medio de un volante giratorio, el foco de luz. El sujeto debe decidir en qué momento deja de percibir la luz. Comienza el experimento y, en un momento determinado, el sujeto da la voz de “alto” para indicar que ha dejado de ver la luz. El experimentador detiene el volante y procede a anotar la distancia. Cuando está a punto de terminar, vuelve a oír la voz del sujeto a través del teléfono: “lo siento, creía que no

veía el punto de luz pero ahora sí lo veo”. El experimentador tacha la anotación y vuelve a girar el volante, alejando aún más el foco del agujero hasta que el sujeto vuelve a dar la voz de “alto”. El experimentador detiene el volante y vuelve a anotar la nueva distancia. Cuando ha terminado, vuelve a oír la voz del sujeto a través del teléfono: “lo siento de nuevo, creía que no la veía pero la vuelvo a ver”. El experimentador, pacientemente, vuelve a girar el volante, vuelve a detenerse al oír la voz de “alto”, vuelve a efectuar la anotación y vuelve a tacharla cuando el sujeto le informa que la luz se le ha vuelto a aparecer. Esto se repite a lo largo de más de media hora. Al cabo de este tiempo la luz se estabiliza.

El sujeto comunica este resultado al experimentador que, por fin, puede anotar la medición correspondiente. El sujeto se relaja y se mueve ligeramente. Entonces la luz desaparece. Contrariado, se comunica con el experimentador y le dice que anule la medición y vuelva a aumentar la intensidad pues ha dejado de percibir la luz. Menea la cabeza con disgusto y, al hacerlo, le parece percibir de nuevo la luz antes de que el experimentador haya tenido tiempo de rectificar la intensidad. Comunica al experimentador que anule la última indicación y vuelve a mover la cabeza, esta vez más despacio, y comprueba que al inclinarla lateralmente con respecto a la línea de visión la luz vuelve a ser percibida. Se vuelve a comunicar con el experimentador pero no consigue establecer contacto (el experimentador se ha ido a su casa).

### **La discriminación de luminancias. Leyes de Weber/Fechner**

Los experimentos para determinar la capacidad de discriminación del ojo de dos áreas de distinta luminosidad, en un rango limitado a unas determinadas condiciones de iluminación y de adaptación, se llevan a cabo de varios modos. Uno de los más corrientes es mostrar al sujeto un campo bipartito provisto de dos pantallas detrás de las cuales están



situadas luces de igual intensidad. Las variaciones de intensidad entre ambos campos se computan con técnicas diversas, que se pueden encontrar en la literatura especializada, pero que consisten básicamente en ir graduando un mecanismo de control hasta que aparezca una diferencia perceptible.

En 1834, el físico Ernst Weber, a quien se considera fundador de la psicología experimental, hizo la observación de que los juicios acerca de la intensidad de una fuente de luz eran siempre juicios relativos y que, por otra parte, la finura de la discriminación era proporcional a la magnitud. Dicho de otro modo: cualquier persona aprecia un cambio en la intensidad de iluminación si en lugar de una vela se utilizan dos velas, pero este cambio es inapreciable si se añade una vela a cincuenta velas. Los resultados de Weber se pueden condensar en que la relación  $dL/L$ , siendo  $dL$  la diferencia o incremento de luminancia y  $L$  la luminancia inicial, es constante. Esto es que

$$dL / L = k$$

Hay muchos modos sencillos de comprobar de modo natural la validez de la ley de Weber. Por ejemplo, si en un día nublado miramos a hacia una nube blanca que destaque lige-

ramente sobre el fondo y luego miramos la misma nube con gafas de Sol, la nube seguirá destacando del fondo sin grandes variaciones, lo que prueba que nuestro sistema visual no distingue *diferencias* entre intensidades, sino *relaciones* entre intensidades. Esto nos permite reconocer y distinguir los objetos aunque las condiciones de iluminación sean muy distintas.

Sin embargo, esta ley no se mantiene de modo uniforme. Cuando la luz es muy débil muchos objetos que se distinguían claramente con luz más intensa se confunden en una masa oscura. Esto es notorio si contemplamos un paisaje conocido a la luz del Sol y a la luz de la Luna. En este último caso, las zonas que no quedan iluminadas directamente por la Luna aparecen uniformemente oscuras aunque resultasen distintas a la luz del día.

Las observaciones de Weber fueron desarrolladas por Gustave Fechner que publicó, en 1860 su obra *Elemente der Psychophysik* (elementos de psicofísica), que incluía el primer método sistemático para medir la percepción del brillo y la formulación de lo que ahora se conoce como la ley Weber-Fechner: “para que la intensidad de una sensación aumente según una progresión aritmética, la intensidad del estímulo debe aumentar según una progresión geométrica”. Fechner fue más allá y planteó varias hipótesis, algunas de las cuales fueron motivo de intensa polémica. Entre otras cosas, Fechner consideró que la diferencia mínima detectable, que se denominaría más adelante JND por sus siglas en inglés (*Just Noticeable Difference*), debía ser considerada la unidad mínima de percepción. Y que la discriminación de luminosidades seguía una ley logarítmica, algo que fue discutido y corregido por S. S. Stevens, y a lo que volveremos más adelante, en el apartado sobre construcción de escalas.

Mediciones experimentales, tales como las efectuadas por König y Brodham (en 1889), Blanchard (en 1918), Lowry (en 1931), Steinhart (en 1936) y otras, recogidas en la obra clásica de Wyszecki y Stiles (1982, véase referencias) han probado la validez del enun-

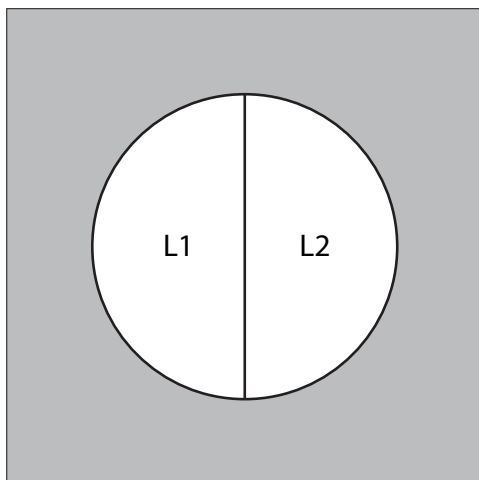


Figura 1.41 Discriminación de intensidades por comparación de dos mitades de un campo bipartito.





ciado más general de esta ley en un rango limitado de valores de intensidad de iluminación. La relación entre aumento de intensidad y percepción de brillo sigue una curva que tiende a volverse horizontal en los extremos y tiene una zona aproximadamente lineal en el medio, si se representan los valores por sus logaritmos, o bien una zona curva que corresponde a una función logarítmica, si se representan los valores normales, en donde se cumpliría estrictamente la ley de Fechner-Weber. Los resultados se muestran en las dos figuras adjuntas, 1.42, 1.43, que resumen la investigación en este campo.

La primera figura, tomada de trabajos del primer cuarto del siglo xx, muestra una distribución general. Esta figura relaciona incrementos diferenciales con luminancias generales de fondo, de color blanco, y cubriendo un área más amplia que la foveal. La discriminación de la luminosidad es difícil de precisar pues depende de las condiciones de adaptación. Es importante tener presente, al observar una curva de estas características, que se están incluyendo en un mismo diagrama condiciones de observación incompatibles como

veremos al hablar de los rangos escotópicos y fotópicos.

Como se observa en la segunda figura, hay una ruptura en la continuidad de la curva para una luminancia de unos  $5 \times 10^{-3} \text{ cd/m}^2$ . Esta ruptura deriva de que, para niveles bajos, la visión foveal (conos, visión fotópica) es reemplazada por la visión parafoveal (bastones, visión escotópica). En el extremo opuesto, para niveles superiores a los  $100 \text{ cd/m}^2$ , la relación  $dL/L$  (denominada corrientemente relación Weber-Fechner), la discriminación alcanza sus niveles menores y permanece constante en torno a 0,01 ( $\log dL/L: -2$ ), es decir, del orden del 1 % de la luminancia del fondo o dicho de otro modo, si los valores de luminancia extremos son 1.000 y 1, se perciben diferencias cuando la relación entre dos valores es más de 1,01. Por ejemplo entre 100 y 101 o entre 900 y 909.

### La cuantificación del contraste

La precisión del concepto y la formulación del contraste son necesarios para comprender adecuadamente todo lo que sigue. Pero tanto el concepto como la formulación pueden re-

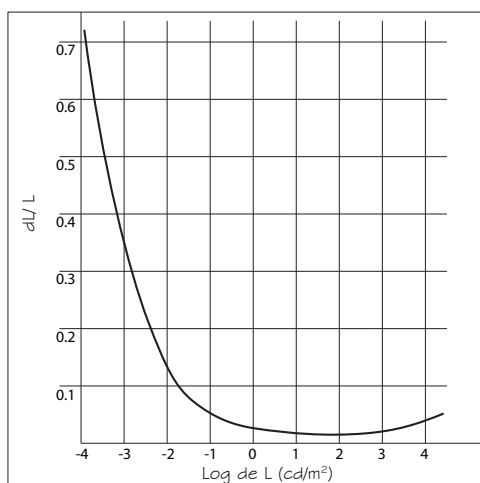


Figura 1.42 Discriminación general de intensidades sobre un rango total no diferenciado. Adaptado de Hecht (1924).

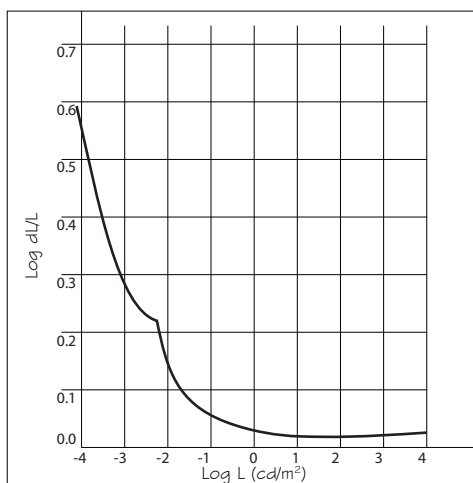


Figura 1.43 Discriminación de intensidades para la visión escotópica y fotópica según datos de la CIE (1924, 1951) (Wyszecki y Stiles, 1982).

sultar confusos en la literatura que circula en la actualidad pues lo que se encuentra en un artículo, un libro o una página web puede ser distinto o contradecir lo que se encuentra en otro.

Aquí, como en otros casos, conviene revisar la historia del concepto y de las formulaciones asociadas. Así nos encontramos con, al menos, tres definiciones bastante distintas de contraste, que han ido evolucionando pero que, desafortunadamente, se encuentran mezcladas en muchos textos sin tener en cuenta esta evolución y el contexto en que se aplican.

El contraste se mide de varios modos. El más corriente, y el que utilizaremos aquí, es el contraste diferencial, el dado por la relación de luminancia entre figura y fondo que deriva de la relación planteada inicialmente por Weber y que puede formularse por la expresión

$$C = (L_1 - L) / L = dL / L$$

donde  $L_1$  es la luminancia de la figura y  $L$ , la luminancia del fondo. En el caso de letras negras sobre papel blanco, este valor suele ser del 80 %. El contraste umbral en condiciones normales (fotópicas) suele ser del 1 % como ya hemos visto más arriba, es decir, que el valor  $C$  sería igual a 0,01. Esta relación se denomina a veces “fracción de Weber”.

Por ejemplo, si queremos saber si el contraste entre un plano oscuro que refleja el 4 % de la luz que recibe situado sobre otro plano que refleja un 5 %, como la luminancia es directamente proporcional a la reflectancia podemos aplicar la fórmula anterior así:

$$C = (0,4 - 0,3) / 0,3 = 0,1 / 0,3 = 0,33$$

El problema de esta definición es que es poco útil para casos más complejos. La segunda formulación fue propuesta por el físico

americano Albert Abraham Michelson (1852-1931, más conocido por sus mediciones de la velocidad de la luz y por el famoso experimento Michelson-Morley) en sus *Studies in Optics* (1927). Esta definición es más adecuada para medir el contraste en una serie de elementos oscuros y claros que se alternan. A veces se la denomina “modulación” y se designa por  $M$ . La fórmula es

$$C = (L_{max} - L_{min}) / (L_{max} + L_{min})$$

donde  $L_{max}$  es la luminancia máxima del conjunto y  $L_{min}$  la luminancia mínima.

Las dos fórmulas anteriores no son equivalentes aunque ambas dan una medida relevante del contraste. Si la diferencia entre las dos luminancias disminuye, el numerador tendería a 0 y el contraste, en consecuencia, también. Y si la diferencia entre ambas aumenta, pongamos a 1.000:1, el cociente sería  $(1.000 - 1) / (1.000 + 1)$  y el contraste tendería a 1.

La tercera definición es más compleja y deriva, entre otras cosas, de los estudios que han puesto de manifiesto la relación del contraste con la frecuencia. El contraste de un enrejado sinusoidal se obtiene por una fórmula similar a la anterior, donde  $L$  es la luminancia relativa (o bien la intensidad o el brillo, según los autores). La luminancia relativa varía de 0 a 1 y la luminancia media es 0,5. El valor de contraste medido (luminancia percibida) es algo inferior debido al efecto de difusión, que aumenta con este tipo de enrejados. El contraste disminuye al aumentar la frecuencia. Esto puede expresarse de modo directo mediante una función denominada función de transferencia de modulación, que se utiliza a menudo en el análisis de la calidad de una imagen.

Tal como puede apreciarse en la figura 1.44 (o en imágenes similares que se pueden encontrar por internet y visualizar mejor en un monitor), al hacerse más alta la frecuencia pero, al mismo tiempo, al disminuir el contraste (hacia arriba en la figura), la sensibilidad al contraste no disminuye por igual sino que se mantiene mejor en las frecuencias medias

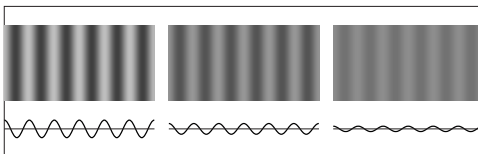


Figura 1.44 Variaciones de contraste para una misma frecuencia espacial.



y disminuye hacia los extremos, con las frecuencias más altas o más bajas.

La función de sensibilidad al contraste, CSF (*contrast sensitivity function*), muestra las variaciones de la agudeza visual en función del contraste. Esta función, que se utiliza en estudios generales sobre la visión, es análoga a la función de transferencia de la modulación, MTF (*modulation transfer function*), que se utiliza en procesamiento de imágenes.

La CSF se define como la respuesta umbral al contraste (siendo la “sensibilidad” el inverso del umbral) en función de la frecuencia espacial expresada en ciclos por grado.

Según la mayoría de los estudios, hay un máximo de sensibilidad o agudeza visual en torno a los 5 ciclos por grado (cpd, *cycles per degree*), que estaría situado alrededor de los 100 cs (*contrast sensitivity*). Como la sensibilidad al contraste es el inverso del umbral, esto corresponde a 0,01 que es el valor que ya habíamos encontrado antes.

Este máximo desciende hacia ambos lados. Por debajo de los 3 cpd baja hasta alcanzar un valor de poco más de 0,1 para 0,1 cpd. Y por encima de los 5 o algo más vuelve a descender hasta alcanzar una sensibilidad 0 hacia los 60 cpd.

### La integración de la resolución y los valores. Funciones de transferencia

Tal como hemos ido viendo a lo largo de los apartados anteriores, no es posible desligar las características geométricas de un sistema óptico de sus características físicas. Las imágenes tienen dos propiedades principales: las que se constituyen en el dominio de la extensión espacial y las que se constituyen en el dominio de la “intensión”, de los valores de intensidad luminosa. Pero ambos están ligados. Y, a menudo, la resolución de una cámara está limitada por las características de la lente más que por la cantidad de píxeles.

Dada la importancia industrial de una calibración exacta de las características de un sistema óptico, en los últimos años se han desarrollado instrumentos para calibrar la capa-

cidad de un dispositivo por medio de funciones que puedan proporcionar una medición objetiva de estas características.

Las ideas en que se fundamentan estos recursos se remontan a finales del siglo XIX pero no se aplicaron de modo general hasta la década de los 1970, aproximadamente. A finales del XIX, las herramientas principales de calibración de instrumentos ópticos se basaban en enrejados de barras negras y blancas que, en el límite de la resolución, tendían a confundirse. Los razonamientos científicos estaban dominados por la óptica geométrica pese a que en algunos aspectos importantes

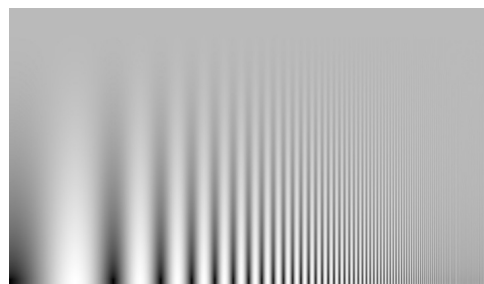
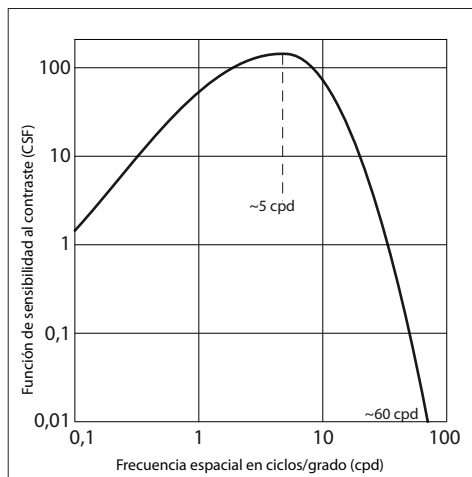


Figura 1.45 Variaciones de la capacidad de discriminación visual al variar el contraste. Arriba: función de sensibilidad al contraste, CSF (Contrast Sensitivity Function). Abajo: imagen equivalente que muestra un patrón sinusoidal cuya frecuencia aumenta hacia la derecha mientras el contraste disminuye hacia arriba. Las frecuencias medias se perciben mejor, con menor contraste, que las bajas y altas.

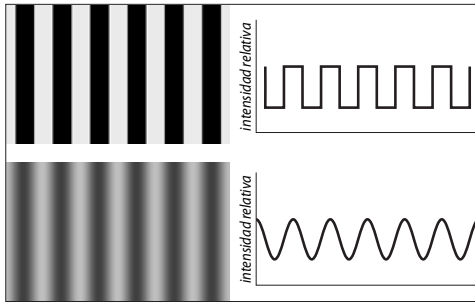


Figura 1.46 MTF. Enrejados de barras y enrejados sinusoidales.

tenían que remitirse a la óptica física. Uno de estos aspectos, que ya hemos visto, era la explicación dada por Airy en 1835 de la formación de los círculos de confusión debido a fenómenos de difracción. La teoría corriente aceptaba que una fuente puntual produciría, sobre el plano de la imagen, un círculo de confusión de tamaño mínimo pero no nulo.

Alrededor de 1940, se comenzaron a utilizar gradientes sinusoidales en lugar de barras. Y, en paralelo, el uso de la transformada de Fourier que había demostrado su gran utilidad en aplicaciones eléctricas, comenzó a aplicarse a otros dominios, como el de la óptica.

ca. Hacia 1950, y en relación con los primeros sistemas de televisión, las imágenes comenzaron a analizarse por métodos de Fourier. Por estas fechas aparecieron en Europa algunas publicaciones notables como, por ejemplo, un libro de P. M. Duffieux, *L'Integral de Fourier et ses Applications a l'Optique* (precedido de diversos artículos iniciados hacia 1935), que contribuirían a difundir estos métodos. En Estados Unidos se publicaron numerosos artículos sobre estos mismos temas hasta que en la década de 1950, el uso de funciones de transferencia óptica (OTF, *optical transfer function*) se fue extendiendo. Se habían comenzado a desarrollar igualmente diversos aparatos de medida, desarrollos que continuaron en la década de 1960 y, en la de 1970, puede decirse que hubo una aceptación general de los nuevos métodos.

Un resultado obvio de este tipo de análisis es que las líneas de muy alta frecuencia serán reducidas en cualquier sistema óptico. En términos corrientes, esto quiere decir ni más ni menos que un borde, contorno o línea de detalle, se transformará en una zona más o menos borrosa, un pequeño gradiente con una zona más o menos intensa en el centro,

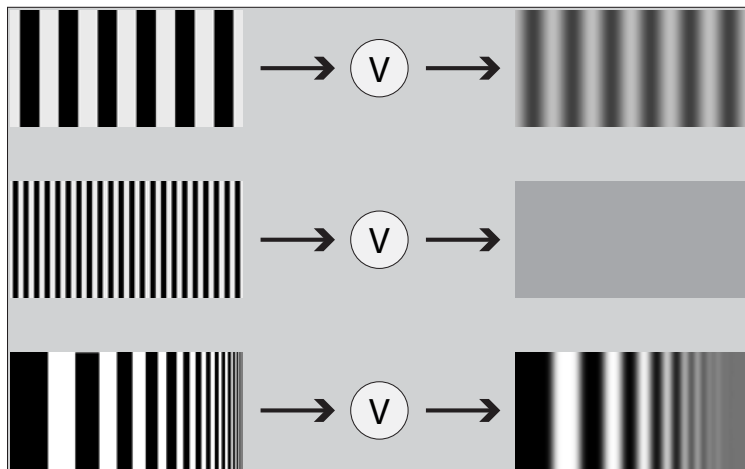


Figura 1.47 MTF. Funciones de transferencia características de un sistema visual: a) Los bordes de enrejados de barras se transforman en bordes desenfocados; b) Si la frecuencia es muy alta el patrón se degrada en un gris uniforme; c) Al aumentar la frecuencia la reproducción se va aproximando al límite de la degradación.



que corresponde al detalle, y un degradado rápido a ambos lados. Cuanto menor sea la calidad del sistema, más notorio será este emborronamiento o desenfoco de los detalles.

Las leyes de Fourier contribuyeron a precisar este análisis. Si se suman sinusoides de diferentes frecuencias se obtienen patrones derivados, y las leyes de Fourier establecen que cualquier patrón ondulatorio puede obtenerse por la suma de una serie de patrones regulares de diferentes frecuencias. Este procedimiento se denomina síntesis de Fourier y la operación inversa, averiguar cuáles son los componentes armónicos fundamentales de un patrón dado, análisis de Fourier. Sin embargo, para reproducir un patrón con esquinas rectangulares, tal como el correspondiente a un borde nítido de una imagen o un patrón de barras negras y blancas, se necesitarían infinitas ondas sinusoidales de frecuencias múltiplos de la inicial ( $\times 3 \times 5 \times 7 \dots$ ) y amplitudes reducidas en la misma proporción ( $\times 1/3 \times 1/5 \times 1/7 \dots$ ), de las que abarcaría el patrón básico. Esto implica que el análisis independiente de los armónicos fundamentales nos dará una medida de la capacidad de un sistema de reproducción para resolver detalles, bordes, contornos de objetos, líneas finas incluidas en la imagen, etc., pues estos detalles se corresponderán con armónicos fundamentales de altas frecuencias.

Supongamos que tenemos una imagen, que denominaremos la “imagen de entrada” (a la izquierda en la figura 1.47), formada por un determinado enrejado sinusoidal como el que se muestra en dicha figura. Contamos con un sistema visual,  $V$ , que puede ser algo tan simple como una lente, que proyecta esta imagen sobre un plano para formar lo que denominaremos la “imagen de salida” (a la derecha, en dicha figura). El resultado será, en principio, una reproducción no exacta, con diversas imperfecciones debidas a las limitaciones del sistema reproductor (aberraciones, imperfecciones diversas). Este resultado se puede representar en un diagrama como el de la figura 1.48, con un punto referido por

un lado, en el eje horizontal, a la frecuencia del enrejado que se da corrientemente en cpd (ciclos por grado) y, por otro lado, en el eje vertical, a la calidad de la reproducción cuyo máximo sería 1 y su mínimo sería 0. En este diagrama se ha representado como límite la curva correspondiente a la difracción, pues, si una lente fuera perfecta, el comportamiento de sus frecuencias más altas dependería tan solo de la difracción. Y esta depende, a su vez, como ya hemos visto, del diámetro de la apertura en relación con la distancia focal (y también de la longitud de onda de la luz incidente, aunque podemos pasar por alto este factor).

Podemos repetir este procedimiento para frecuencias cada vez mayores y, así, iremos obteniendo diferentes calidades de reproducción hasta llegar a un punto en el que el enrejado se confunde en un área gris uniforme que correspondería a un nivel de reproducción 0. De este modo obtendremos una curva que es una representación de la calidad de la lente, de su capacidad para resolver detalles en términos de resolución espacial y contraste de valores.

Esta curva se denomina MTF (*modulation transfer function*), función de transferencia de la modulación. El término alude a la transformación de los componentes modulares básicos, los armónicos fundamentales de una imagen original, que se convierten en otros componentes modulares en la imagen de salida. La enorme importancia de esta herramienta es que, una vez conocida, se puede aplicar a cualquier imagen y podemos, por

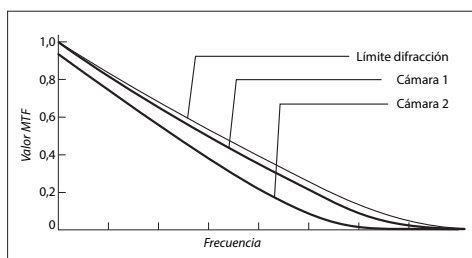


Figura 1.48 MTF. Curvas de transferencia características de lentes de cámaras comparadas con una lente ideal.



tanto, calcular, *a priori*, cuál será la capacidad de resolución de un dispositivo. En una lente o un sistema óptico equivalente, la formulación básica se reduciría a  $MTF(\nu) = \text{salida}(\nu) / \text{entrada}(\nu)$ , donde  $\nu$  es la frecuencia espacial.

Por convención, la MTF se normaliza haciendo que sea igual a 1 para valores de frecuencia cero. Los valores de frecuencia alta, que tienen menor contraste, hacen que el valor de la MTF descienda hasta acercarse a 0, que sería el límite de resolución de un sistema óptico donde las frecuencias altas se fundirían en un gris uniforme. Por otro lado, la MTF puede generarse para los tres canales de color, lo que permite también obtener información exacta sobre la reproducción de los colores de un dispositivo determinado. En este resumen me limitaré a las imágenes en blanco y negro.

La MTF está directamente relacionada con otras funciones características que cito brevemente, pues el lector interesado las encontrará en la literatura sobre este tema.

La PSF (*point spread function*) o “función de dispersión de un punto” es la función que muestra la distribución de intensidades de un punto.

La LSF (*line spread function*) o “función de dispersión de una línea” es la función que muestra la distribución de intensidades de una línea o una hendidura. Es una medida de la capacidad de un sistema para formar imágenes nítidas. La OTF, a la que me referiré más abajo, es de hecho la transformada de Fourier de la LSF.

La PTF (*phase transfer function*) o “función de transferencia de fase” es una medida de la fase relativa de la imagen en función de la frecuencia. Un cambio de fase de  $180^\circ$ , por ejemplo, indica que se han alternado franjas blancas con franjas negras lo que equivaldría a una inversión en negativo de la imagen. En la mayoría de los casos este efecto puede obviarse pero hay otro en que no. Una lente imperfecta o un sistema en el que la luz cambie con el tiempo pueden producir desplazamientos de fase. En el primer caso se traduciría en

desviaciones con respecto a un eje óptico que pueden ser irrelevantes si toda la imagen se desplaza y en segundo en desviaciones con respecto a la secuencia prevista. Tiene menor interés práctico pero es necesario citarla pues está implicada en la OTF.

La OTF (*optical transfer function*) o “función de transferencia óptica” de un sistema de captación de imágenes es una medida real de cómo responde el sistema a una fuente real conocida. Es una medida de la resolución, entendida no solo como el número de píxeles de la imagen resultante, sino como la capacidad de resolver adecuadamente frecuencias altas, es decir, zonas de transición rápidas (bordes, líneas) y, por tanto, de reproducir adecuadamente los detalles, que no dependen solamente del número de píxeles sino de esta capacidad de mantener el contraste de valores en zonas críticas. La OTF se compone de dos componentes: la MTF y la PTF. El término que refleja la magnitud es la MTF y, en general, este es el dato principal de interés.

## 1.9 Rangos dinámicos. Escalas de valores

### Umbral absolutos y relativos

*Rangos absolutos. Umbral inferiores.* La detección de luminosidades mínimas se lleva a cabo en condiciones de plena adaptación a la oscuridad. En todos los casos, la adaptación a la oscuridad para los sujetos del experimento es de  $\frac{1}{2}$  hora a 1 hora. Si han estado previamente en ambientes muy luminosos, el tiempo de adaptación debe ser aún mayor. No voy a entrar en el detalle de las diferentes variantes, pero merece la pena recordar que para un análisis más preciso deben recrearse estas condiciones, que pueden encontrarse en la literatura especializada (estímulo presentado en exposiciones largas y sin restricciones de ángulo, con ángulos más restringidos, de hasta  $1^\circ$ , con tiempos de exposición más restringidos, con una luz de 507 nm, en



la cual la eficacia es máxima en condiciones escotópicas o con luces de otras longitudes de onda dominante, etc.).

Los trabajos clásicos de Hecht, Shlaer y Pirenne, entre otros, permiten fijar como cantidad de referencia, para ojos adaptados a la oscuridad, y para una luz de unos 510 nm, la cifra de unos  $10^{-10}$  cd/m<sup>2</sup>. Pero hay algunos aspectos que conviene comentar con algo más de extensión.

La referencia fundamental para fijar los umbrales mínimos es el trabajo presentado por Selig Hecht, Simon Shlaer y Henri Pirenne en 1942 (véase Hecht *et al.*, 1942). En este artículo, los autores comienzan por revisar los principales trabajos similares realizados hasta la fecha (el primero de los cuales, por Langley, se remonta a 1889) señalando sus errores o sus limitaciones (con algunos resultados valiosos y cercanos a los suyos, como los llevados a cabo por von Kries y Eyster en 1907). Citan también las mediciones de las magnitudes de las estrellas, a las cuales me referiré más adelante. Y pasan a describir el complejo aparato utilizado para sus mediciones, lo que lleva bastantes páginas. En síntesis, se trata de un aparato que, en primer lugar, filtra bandas del espectro para que la luz incidente sea de 510 nm, la frecuencia a la que, como también hemos visto, el ojo es más sensible en condiciones escotópicas, y luego hace pasar la luz por una serie de dispositivos y lentes para asegurar que es recibida correctamente y con el tamaño deseado. El dispositivo incluía también un mecanismo de obturación y una serie de dispositivos de medición muy sofisticados. Las condiciones de observación pasan por al menos 30 minutos durante los cuales el sujeto debe permanecer en oscuridad total y con un ángulo general de observación de unos 20° medidos desde la fóvea pues, tal como ya hemos visto, en estas condiciones los bastones son mucho más sensibles que los conos y la mayor densidad de bastones comienza a unos 18°, con máxima sensibilidad a la luz entre los 20° y los 30°. El ámbito del estímulo observado era de

10' de diámetro, y las exposiciones de 0,001 segundos.

Las conclusiones del experimento fueron las siguientes. La energía emitida captada por el sujeto del experimento fue de 2,1 a  $5,7 \times 10^{-10}$  erg en la córnea, lo que equivale a entre 58 y 148 cuantos de luz. Estos valores deben ser reducidos debido a la reflexión de la córnea (4 %), la absorción debida a los medios oculares (en torno al 50 %), y la transmisión retiniana (80 %). El resultado es que entre 5 a 14 cuantos llegan a los bastones. Pero en la zona correspondiente a los 10' de ámbito, había unos 500 bastones, por lo que un análisis estadístico llevó a la conclusión que, en el umbral de visión, bastarían del orden de 5 a 8 cuantos para producir una respuesta. Esto es coherente con la idea intuitiva de que, en teoría, bastaría con que un único cuanto llegase a afectar a un receptor retiniano para que se generase una reacción en el sistema visual.

Es importante subrayar que en este tipo de experimentos hay un grado de fluctuación que depende tanto del sujeto como del estímulo. Cada bastón contiene del orden de 4 millones de moléculas de pigmentos que pueden activar las señales enviadas por los bastones. Estas moléculas pueden cambiar de estado espontáneamente o por efecto de estados subjetivos del sujeto del experimento. Es bien conocido que podemos ver pequeñas luces con los ojos cerrados por diversas causas, o que si nos levantamos durante la noche y miramos una pared blanca veremos pequeños puntos erráticos, cierto nivel, casi imperceptible, de "ruido visual". Por otro lado, el estímulo también puede fluctuar y por esta razón los resultados de los experimentos se dan como medias estadísticas de múltiples observaciones. Sin embargo, una de las conclusiones del trabajo de Hecht *et al.*, es que en experimentos anteriores se había tendido a sobrevalorar las fluctuaciones subjetivas frente a las objetivas.

Como todo esto es muy abstracto, resultará más interesante relacionar estos datos con otro tipo de referencias. Y la mejor referencia es, sin duda, la intensidad de las estre-





llas, que ha sido analizada y clasificada por astrónomos desde hace muchos siglos. Pero también por marinos, cuya vida dependía de su capacidad para reconocerlas.

Las estrellas se clasifican por su “magnitud”. Esta clasificación, que se sigue utilizando, viene del astrónomo griego Hiparco que por medio de escrupulosas mediciones de cerca de 1.000 estrellas (en Rodas, entre 161 y 127 aC), clasificó las estrellas en 6 categorías según su brillo aparente a simple vista, de la más brillante (magnitud 1) a la apenas perceptible (magnitud 6). Esta clasificación fue adoptada por Ptomoleo, unos 400 años más tarde, y por todos los astrónomos occidentales que le siguieron hasta finales del siglo XIX, en que se introdujo otra clasificación, por tipos espectrales, conocida como la “clasificación espectral de Harvard” (en 1890). Pero la clasificación clásica se sigue utilizando sobre todo después de que en 1856 el astrónomo británico Norman Pogson (1829-1891) propusiera un método más riguroso que fue adoptado por todos los astrónomos. Este método parte del principio de que una estrella de magnitud 1 es 100 veces más brillante que una de magnitud 6. Las magnitudes intermedias se consideran que aumentan según la raíz quinta de 100 (2.512...). Dicho de otro modo: propuso adoptar una escala logarítmica con un factor de 0,4 (el logaritmo de 2.512...).

En la actualidad, se utiliza un sistema de medición que se basa en estos ilustres predecesores pero de un modo más sistemático. Se define una diferencia de cinco magnitudes como igual a una relación de brillo de 100, y se toma como referencia de la escala una estrella concreta, la estrella Vega, a la que se asigna una magnitud de 0 (mediciones recientes han modificado este valor a 0,03) y, a diferencia de las escalas de Hiparco, las escalas modernas no ponen un límite superior. La relación entre las intensidades o “magnitudes” de dos estrellas, viene dada por la fórmula siguiente:

$$m_1 - m_2 = -2,5 \log_{10} (F_1 / F_2)$$

en donde  $m_1$  y  $m_2$  son las magnitudes (intensidades) de las dos estrellas,  $F_1$  y  $F_2$ , sus flujos lumínicos y 2,5 es aproximadamente la raíz quinta de 10. No hay que perder de vista que en la escala de Hiparco, como en la nuestra, los valores menores designaban intensidades mayores. De ahí que las mediciones actuales hayan dado como resultado luminosidades mayores a las estimadas por Hiparco y hay unas cuantas estrellas con un valor negativo. Si el Sol estuviera situado a una distancia de 10 parsecs de la Tierra, su magnitud sería de 4,83. Pero a la distancia a la que está de nosotros, su magnitud, en la escala de Hiparco corregida, sería de -26,8.

Con esta clasificación se ha extendido la clasificación original para incluir otras estrellas que se salían de la lista, como Sirio, la más brillante, considerada de magnitud -1,47 o Canopus, la segunda más brillante, de magnitud -0,72. Otras estrellas notables son Vega (0,03), Betelgeuse (0,58), Espiga (1,04), Pólux (1,15). La extrapolación de esta escala permite también incluir a planetas como Marte (-2,8) o Venus (-4,4). O la Luna llena (-12,6). O el Sol (-26,8). También por el otro extremo se han incluido objetos observables con la ayuda de telescopios, como el quasar más brillante (+12,6) o los últimos objetos detectables por el telescopio espacial Hubble (+30).

Hay que tener en cuenta que la observación puede venir alterada por el hecho de que las estrellas se nos aparecen como puntos irregulares de los que surgen rayos irregulares divergentes. Los rayos regulares con los que se representan convencionalmente las estrellas no corresponden a la realidad. Los rayos son más largos y más irregulares cuando el ojo está rodeado de oscuridad y la pupila bien abierta. Si se mira a una estrella a través de un agujero de 1 mm, para minimizar las irregularidades de nuestro ojo el disco aparecerá más regular, pero si lo desplazamos en torno a la pupila se revelarán estas irregularidades que dependen de nuestro propio ojo. Estos efectos varían con cada persona y también dependen de otros factores como, por ejemplo, la desviación de los rayos debida



a las pestañas, un efecto que se intensifica si entrecerramos los ojos.

Las estrellas “corrientes”, visibles desde una ciudad con cielo despejado, tienen magnitud 3. En el campo y en buenas condiciones de visibilidad, la mayoría de la gente puede ver estrellas de magnitud 6 y bastantes personas pueden ver estrellas de magnitud 7 si las condiciones de observación son muy buenas (lejos de luces urbanas y con un cielo despejado y buena vista). Dado que la sensibilidad lumínica de los bastones es muy superior a la de los conos, la percepción aumenta si se mira algo de reojo a las estrellas.

En la dirección del horizonte vemos menos estrellas. Esto es debido a que los rayos de luz tienen que atravesar una parte mayor de la atmósfera, lo que disminuye su intensidad debido principalmente al fenómeno de dispersión que ya se ha comentado.

Según Minnaert (Minnaert, 1954, párrafo 47), si se escoge la estrella Capella, con una magnitud de 0,3 en la escala de magnitudes aparentes, deberíamos situar una vela (una intensidad de 1 cd) a una distancia de 900 metros para obtener la misma intensidad. La intensidad recibida, en luxes, supondría  $1/900^2 = 1/81.000 \approx 0,00001$  luxes. La intensidad recibida sobre una superficie iluminada por la Luna llena es del orden de los 0,2 luxes, la misma que produciría una bombilla que emitiese 100 cd a algo más de 20 metros de distancia ( $100/20^2 = 100/400 = 0,25$ ).

Otros datos que pueden mencionarse provienen de la navegación marítima. Para señales luminosas (faros, etc.) se da la cifra mínima de 0,5 *photopic mile-candle* (1 *mile-candle* es la iluminancia producida por una fuente puntual de 1 candela fotópica de intensidad, a 1 milla marítima de distancia en una atmósfera despejada).

Estos valores son debidos a que la atmósfera no es perfectamente transparente. En estos casos, la iluminación recibida se obtiene por la fórmula  $E = It_D / D^2$ , donde  $I$  es la intensidad de la fuente,  $t_D$  la transmisividad de la atmósfera por unidad de distancia y  $D$  la distancia. Los valores característicos de  $t$  son

0,88 o superior (despejado,  $v$ , visibilidad de más de 31 millas), 0,73 a 0,88 (muy claro,  $v$  de 12 a 31 millas), 0,53 a 0,73 (claro,  $v$  de 6,2 a 12), 0,21 a 0,53 (leve niebla,  $v$  de 2,5 a 6,2), 0,004 a 0,21 (niebla,  $v$  de 1,2 a 2,5).

**Rangos absolutos. Umbrales superiores.** La respuesta a la pregunta ¿cuál es la intensidad más intensa que podemos percibir? carece propiamente de sentido pues cualquier intensidad por encima de un cierto valor que vemos como “blanco” será percibida igualmente como “blanco”. Hay que reconvertirla en otras dos: ¿qué valor de intensidad tiene lo que denominamos “blanco”? y ¿a partir de qué valores se produce una irradiación en lo blanco que acaba en el deslumbramiento o, si se prefiere, ¿a qué llamamos “deslumbramiento”.

**Rangos absolutos. Entornos de referencia.** He indicado cuáles serían los umbrales absolutos, inferiores y superiores, las cantidades mínimas y máximas que el sistema visual humano es capaz de percibir, prescindiendo de las condiciones de adaptación. Pero en la práctica interesa más saber cuáles son los umbrales inferiores y superiores que percibimos en circunstancias concretas, en unas determinadas condiciones de adaptación. Es obvio, tras todo lo dicho, que no puede haber una respuesta sencilla a esta pregunta pues estas circunstancias son demasiado variables. Pero podemos acercarnos a la respuesta recordando los diferentes modos en que trabaja el sistema visual en circunstancias características.

Cuando la intensidad global es muy débil, como ocurre de noche, la percepción de la luz corre principalmente a partir de los bastones y la respuesta es distinta a cuando la intensidad es “normal” (diurna). En el primer caso, se habla de condiciones escotópicas y en el segundo, de condiciones fotópicas. Entre ambas se sitúan unas circunstancias más complejas, denominadas mesópicas. Las diferentes condiciones de adaptación se clasifican en estos tres grupos.

El rango *escotópico* (del griego *skotos*, “oscuridad”) abarca desde  $10^{-6}$  a  $10^{-2}$  cd/



$\text{m}^2$ , aproximadamente. En estas condiciones, la pupila está ampliamente dilatada (7,5 a 8 mm) y los conos no participan de modo pleno en la percepción.

El rango *mesópico* comienza hacia los  $10^{-3} \text{ cd/m}^2$  aproximadamente y termina en una zona variable, que depende de la composición espectral del estímulo y que estaría en torno a 1 o  $10 \text{ cd/m}^2$  (para más detalles, véase Wyszecki y Stiles, *op.cit.*, p 406).

El rango *otópico* (del griego *photos*, “luz”) abarca desde 1 o 10 hasta  $10^4$  o  $10^8$ , en que se pueden producir daños en el sistema visual: no es posible, como hemos visto, fijar un límite superior. La pupila está contraída, con una apertura de entre 2,2 y 3 mm, que se considera el tamaño óptimo para minimizar la difracción y las aberraciones del sistema dióptrico. La agudeza visual aumenta a medida que aumenta la iluminación hasta llegar a un punto en el cual comienza a descender debido al deslumbramiento. Todo esto se tratará con mayor detalle en los capítulos siguientes.

El rango relativo a las condiciones más habituales (luz diurna, visión otópica) abarcaría, por tanto, alrededor de  $10^8$  valores de luminancia.

El rango absoluto global abarcaría al menos  $10^{10}$  valores de luminancia relativamente corrientes (de la luz de las estrellas en una noche oscura a la luz del Sol al mediodía) y hasta  $10^{14}$  valores de luminancia en casos extremos (luces muy débiles y luces muy intensas vistas directamente).

Debe también tenerse en cuenta que la luminancia recibida por el ojo no depende exclusivamente de la luminancia de la fuente sino también del tamaño de la retina. Por esta razón, a partir de los trabajos del bioquímico y psicólogo americano Leonard Troland (1889-1932) se introdujo la unidad que lleva su nombre, el *troland*, que corrige los valores de luminancia según el tamaño de la zona de la retina afectada. La luminancia medida en trolands se obtiene multiplicando la luminancia de la fuente por el área de la pupila en  $\text{mm}^2$  ( $T = L \times p$ , siendo  $L$  la luminancia

en  $\text{cd/m}^2$  y  $p$  el área de la pupila en  $\text{mm}^2$ ). Un *troland* se define como la luminancia retiniana producida por  $1 \text{ cd/m}^2$  para un área de  $1 \text{ mm}^2$ .

La figura 1.49 resume los rangos principales que se han citado hasta aquí, incluyendo los diámetros de pupila para las diferentes condiciones de adaptación y los valores correspondientes en trolands. El quiebro en la lista de valores corresponde a una distinción que también se aplica en ocasiones entre trolands correspondientes a condiciones otópicas y escotópicas.

Estos rangos nos permiten aproximarnos mejor a la pregunta hecha al comienzo de este apartado, sobre cuáles son los umbrales extremos de situaciones características. Pero son aún distinciones teóricas, aunque las indicaciones son muy valiosas pues nos hacen ver cómo el predominio de uno u otro tipo de receptor afecta las características de la visión: a la percepción de los colores y a la agudeza visual, principalmente, que se modifican notablemente al pasar del rango otópico al escotópico (la percepción de los colores se desplaza hacia los azules, que se distinguen bien, cosa que deja de ocurrir con los colores anaranjados y rojos, y la agudeza visual disminuye).

Se puede reformular la pregunta de este modo: ¿cuáles son las condiciones “normales” de iluminación ambiental?

Podemos considerar el caso de un espacio interior con “buena iluminación”, dicho con mayor precisión, con luminancias comprendidas entre 1 y  $1.000 \text{ cd/m}^2$ . En estas condiciones, el ojo puede pasar de una región escasamente iluminada, a una región muy iluminada sin que el tiempo de adaptación sea determinante. Dadas estas condiciones que, como veremos, no dejan de ser bastante artificiales, otra pregunta importante es: ¿cuántos intervalos diferenciales mínimos podría distinguir un sujeto dado? Dicho de otro modo, ¿cuántos escalones podría tener una escala de grises situada en una zona de iluminación media de este espacio sin que se confundieran unos con otros?

## Rangos dinámicos

Como acabamos de ver, el rango de luminancias que el sistema visual puede percibir es enormemente amplio. Pero varía en cada caso según las diferentes escenas y las condiciones de adaptación bajo las que observemos una escena. En cada caso tendremos un diferente “rango dinámico”, que se define como la proporción entre el máximo y mínimo de un valor variable. El término *dinámico* implica que podemos percibir diferentes valores si cambiamos nuestro punto de vista sin modificar excesivamente las condiciones de adaptación.

Los sentidos de un ser humano abarcan un rango dinámico muy amplio si se tienen en cuenta todos los niveles de adaptación. El oído abarca un rango de hasta 100 dB, que iría desde la percepción de leves roces en una habitación insonorizada hasta la asistencia a un concierto de rock con los altavoces a plena potencia. Como podemos traducir fácilmente los valores anteriores a decibelios resulta que la vista humana abarca, como ya hemos visto, un rango equivalente a unos 80 dB en condiciones fotópicas ( $10^8$ ), si bien en

condiciones corrientes de adaptación este rango es bastante inferior y se situaría en torno a los 30, 40 o 50 dB, una cifra que vale también, aproximadamente, para la percepción auditiva (una cinta magnética tiene un rango dinámico de unos 55 dB y un disco de vinilo de unos 65 dB).

Recordemos, antes de seguir, que el *decibelio* es una unidad de medida que se emplea para expresar cómodamente la magnitud de una cantidad física, generalmente energía o intensidad, en relación con un nivel de referencia determinado (implicado o directo). Tiene la ventaja de que permite representar números muy grandes o muy pequeños, que se distribuye según una escala logarítmica, que se corresponde aproximadamente con la percepción humana, tanto auditiva como visual, y que permite simplificar los cálculos mediante adiciones y sustracciones en lugar de multiplicaciones y divisiones. Se desarrolló por los ingenieros de los Bell Telephone Laboratories (de ahí el nombre) y se le dio el nombre actual hacia 1923.

Si  $L_{dB}$  representa la proporción entre dos potencias  $P_1$  y  $P_0$ , su expresión en decibelios se calcula mediante la fórmula:  $L_{dB} = 10 \log_{10}$

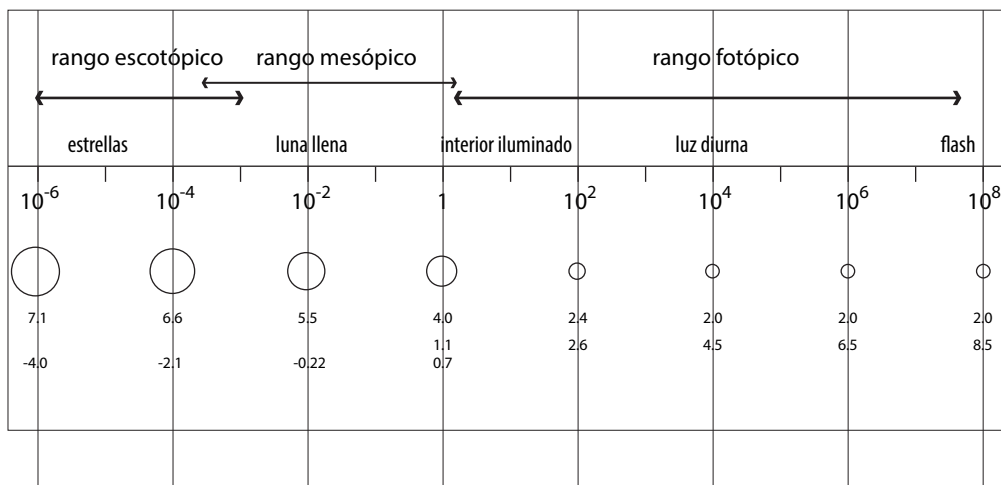


Figura 1.49 Rangos visuales absolutos. La fila de valores  $10^{-6}$ ,  $10^{-4}$ , etc., indica luminancias; los círculos, diámetros de pupila; la fila por debajo de los círculos, sus diámetros en mm, y la siguiente fila, valores equivalentes en trolands.



$(P_1 / P_0)$ . Así si, por ejemplo, queremos expresar en dB un rango dinámico de 1.000:1, todo lo que tendremos que hacer es substituir estos valores en la fórmula anterior, lo que nos dará  $L_{dB} = 10 \log_{10} (1.000/1) = 10 \times 3 = 30$ . La tabla de la figura 1.50 muestra ejemplos de rangos dinámicos expresados en dB y de otros modos.

El rango dinámico se da también en términos del *factor de contraste*. El contraste se define corrientemente, como ya hemos visto, a partir de la relación entre dos valores de luminancia. Y el término factor de contraste se utiliza generalmente para indicar la relación entre la luminancia máxima y mínima que puede darse en una determinada escena o en un determinado dispositivo.

En circunstancias óptimas, la visión humana opera con un rango dinámico de 30 a 40 dB, lo que equivale a un rango de luminancias y, por tanto, a un factor de contraste de 1.000 a 1 o de 10.000 a 1. En circunstancias “corrientes”, esto disminuye a unos 20 dB o un factor de contraste de 100:1. Las condiciones habituales de trabajo en una oficina no muy bien iluminada hacen que el factor de contraste pueda llegar a ser tan bajo como 5:1.

Es importante tener en cuenta que aunque el ojo pueda captar alrededor de 5 órdenes de magnitud simultáneamente, en casos extremos, y podamos percibir diferentes zonas de una escena de estas características, el rango que podemos percibir en una zona limitada a ángulos de visión más pequeños es mucho

menor. Aunque hay diferencias entre los resultados experimentales, la mayoría de los investigadores aceptan que este rango efectivo está alrededor de 150:1 (Vos, 1984). Esta limitación es debida a los efectos de dispersión que se producen en el ojo. Trabajos que se remontan a mediados del siglo xx (Moon y Spencer, 1945) mostraron que cualquier borde con alto contraste dispersa al menos un 4 % de la energía en la retina hacia la zona oscura, reduciendo la visibilidad del borde en unos pocos grados. Esta dispersión aumenta con el contraste. Para un contraste de 25:1 se reduce en un factor de 2 en la región oscura y, para un contraste de 150:1, por un factor de 12. Esto no quiere decir que no percibamos estos contrastes en escenas reales, sino que nuestra capacidad efectiva para discriminar detalles en este tipo de zonas se reduce debido al fenómeno del resplandor local.

Es importante esta observación, entre otras cosas para evitar equívocos sobre los enormes contrastes dinámicos que se ofrecen comercialmente. En la mayoría de las circunstancias podemos asumir con seguridad 100:1 como un factor de contraste de referencia suficientemente bueno, lo que quiere decir que, a partir de los experimentos citados más arriba, seremos capaces de diferenciar entre dos zonas si su relación de intensidades es del orden de 1,01 (un contraste de 1 %). En el capítulo siguiente veremos cómo se traducen estos valores en términos digitales y de calibración de dispositivos virtuales.

Situación	Factor de contraste	Rango (dB)
Gama total	100.000.000.000.000:1	140
Gama fotópica	100.000.000:1	80
Exterior al Sol	100.000:1	50
Interior iluminado	10.000:1	40
Ojo adaptado	10.000:1	40
Negativo analógico	10.000:1	40
Monitor gama alta	1.000:1	30
Sensor DSLR	500:1	27
Monitor LCD	300:1	25
Monitor CRT	150:1	22
Impresión papel, calidad fotográfica	100:1	20
Impresión papel, corriente	50:1	17

Figura 1.50 Rango dinámico de diferentes escenarios. El factor de contraste (segunda columna) indica los valores extremos de luminancias en  $\text{cd/m}^2$ .



## La construcción de escalas de valores

La siguiente pregunta que cabe plantear es ¿cómo construimos escalas de valores que nos permitan relacionar una determinada escena con una determinada ordenación de valores?

Para que se comprenda mejor el complejo sentido de la pregunta anterior preguntémosnos: ¿cuál es el mínimo número de luminancias que nos permiten distinguir figuras de fondos? La respuesta en este caso es obvia: dos. Es el grado de contraste mínimo que necesitamos en una escena para poder distinguir objetos.

Ahora bien, en esta respuesta se están implicando dos puntos de partida posibles bien distintos. Que la escala de luminosidades sea “dos” puede significar dos cosas muy distintas: o bien que el número de objetos con diferente luminosidad sea mínimo en un campo amplio o bien que el número de objetos con diferente luminosidad sea máximo, aunque la amplitud del campo luminoso sea mínima.

Es decir, podemos considerar dos situaciones posibles. La primera puede venir ejemplificada por una gran sala, iluminada con un gran foco, muy potente, y repleta de cubos blancos y negros situados a igual distancia de este gran foco y del observador, que podría estar situado en una posición coincidente con la del foco. La segunda, por una gran sala, repleta de cubos de más de cien matices diferentes de gris, pero iluminada con una única luz tan débil que solo permitiría apreciar dos tipos de intensidades.

A partir de estas matizaciones iniciales nos podemos preguntar cuántos valores podemos *percibir* y si estos valores se pueden ordenar, esto es, si podemos construir una escala de grises de un modo sistemático.

Tampoco es fácil responder directamente a estas dos preguntas. La razón principal para esta dificultad es que la percepción de las diferencias de intensidad se produce localmente mediante una asignación intuitiva que depende de los límites de los valores comparados. Es decir, dada una escala de grises y

un nuevo valor de gris, cualquier observador será capaz de situar este nuevo gris en la escala. Pero si se juzga un determinado gris A como igual a otro determinado gris B y luego se juzga que este gris B es igual a otro gris C, puede ocurrir que al comparar directamente el gris A con el gris C, yuxtaponiéndolos, se observe una diferencia que antes había pasado inadvertida.

Dicho de otro modo: una cosa es hablar de cuántos grises “reconocemos” en una escena, en el sentido de relacionarlos con una escala de referencia (como el sistema de zonas de que hablaremos más adelante), y otra cosa es hablar de cuántos grises “distinguimos” si se colocan uno al lado de otro.

Los experimentos centrados en averiguar la capacidad de discriminación local, haciendo variar un punto luminoso en un campo de intensidad luminosa constante, han mostrado que la capacidad media para *reconocer* diferentes niveles de grises en una escena sería de unos 20 grises para un observador medio, aunque algunos de estos estudios dan el valor de hasta 40 (véase, por ejemplo, Marion, 1987, p. 182 de la edición inglesa de 1991) y otros lo reducen a 10 o 15 (por ejemplo Pratt, 1978, p. 136 de la reedición de 2007). Es una cifra difícil de precisar pues varía mucho con las personas y también con las condiciones de experimentación. Un observador experimentado puede distinguir y nombrar (relacionar con una determinada escala de referencia) un número mucho mayor de grises que otro. Igual que un sumiller apreciará múltiples matices en un vino que se le escapan a alguien menos acostumbrado a las catas.

Pero la capacidad de *distinguir* estos valores cuando se yuxtaponen es mucho mayor. En condiciones óptimas, pueden ser necesarios hasta 200 valores para que un degradado aparezca como continuo (es decir, para que no se distingan los diferentes grises que realmente lo componen) debido al efecto denominado bandas de Mach, que se describirá más adelante.

Esta introducción nos servirá también para hacer más comprensible la razón por



la cual los científicos distinguen al menos cuatro tipos de escalas que son relevantes en diferentes casos. Estos tipos fueron propuestos por el psicólogo americano Stanley Smith Stevens (1906-1973), en 1946, y en el contexto de una discusión sobre las formulaciones de Fechner a las que volveremos en seguida.

Una escala *nominal* es una escala útil en ciertos casos pero más bien trivial. Una escala de este tipo define de modo más bien arbitrario una serie de nombres y establece una relación entre los nombres y las muestras. Así, podríamos construir una escala con nombres tales como negro, gris oscuro, gris claro y blanco. O bien podríamos utilizar nombres convenidos como gris “marengo”, “ceniza”, “humo” o “perla”, sin preocuparnos ni tan siquiera por su ordenación. Estas escalas son muy útiles en la práctica pero no presentan ninguna dificultad de construcción.

Una escala *ordinal* es una variante más elaborada que la anterior, donde los diferentes elementos están ordenados de un modo estricto. El ejemplo anterior podría desarro-

llarse, introduciendo nombres más elaborados que hicieran la escala fácilmente extensible y proporcionando un orden intuible aunque sin entrar en la cuestión de la regularidad de este orden.

Una escala de *intervalos* es otra variante más elaborada que las anteriores, donde se parte de algún tipo de método que garantice la regularidad de los intervalos. Las escalas utilizadas en los termómetros, o en las cintas métricas o en los calendarios, son ejemplos obvios de este tipo de escalas. En una escala de este tipo las diferencias entre dos muestras puede ser medida mediante números y pueden llevarse a cabo operaciones de sumas y diferencias.

Antes de mencionar la cuarta y última escala precisemos que las escalas ordinales pueden partir de un valor arbitrario o pueden tener un origen que se corresponda con algo real. Un ejemplo de esto último sería una escala de temperaturas cuyo origen fuera el 0 absoluto ( $-273^{\circ}$ ). Otro ejemplo adecuado es el de escalas, como las de luminosidad, que cuentan con un punto al que puede asignarse

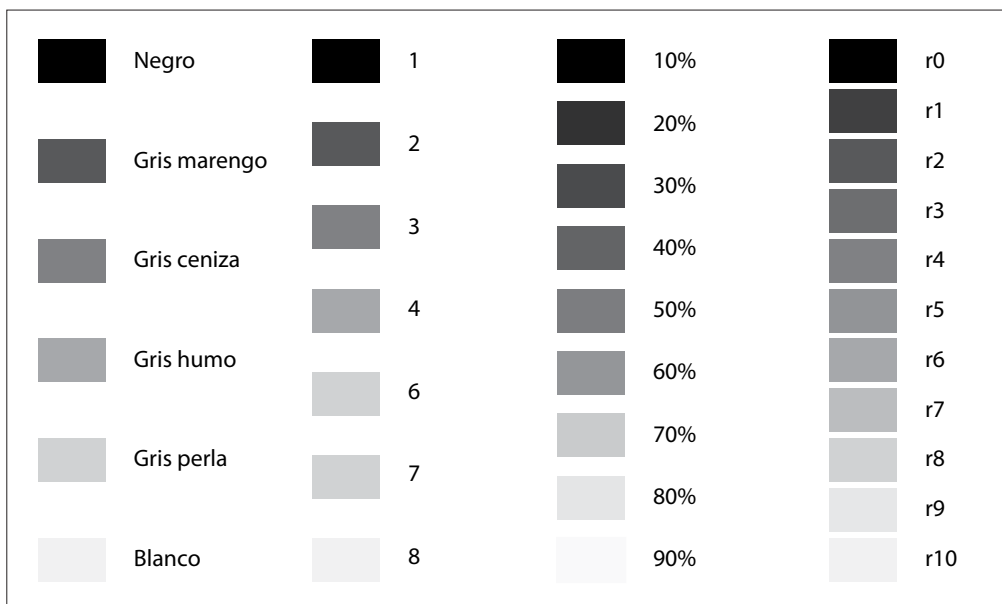


Figura 1.51 Escalas a) Nominal, b) Ordinal, c) De Intervalos, d) De Relaciones.





el valor 0, de tal modo que se corresponda con un valor comprensible (negro).

Una escala de *relaciones* o escala *proporcional* tiene todas las propiedades de las anteriores pero cuenta por añadidura con un origen natural y un método para relacionar los diferentes niveles entre sí. Esto implica que pueden darse relaciones con sentido entre los elementos de la escala, tales como que un elemento es el doble, o el triple, o la mitad o la tercera parte de otro. La mayoría de las escalas utilizadas corrientemente en el mundo de la ciencia son de este tipo. La gran dificultad, sin embargo, es que, en muchos casos, no es posible especificar un origen natural.

Resulta claro por consiguiente que la generación de una escala de relaciones requiere de alguna formulación coherente que abarque el fenómeno que se quiere medir.

El primero en intentar esta formulación fue Fechner, que partió de la fórmula de Weber que ya hemos visto y que puede reescribirse como sigue:

$$dS/(S1 + S2) = k$$

donde  $S1$  y  $S2$  son dos estímulos que pueden ser resueltos por el observador, esto es, que pueden ser percibidos como mínimamente distintos. Y partiendo de esta ley, en un rango determinado y tomando  $s$  como unidad de sensación correspondiente al estímulo  $x$  e integrando en todo el rango, con  $a$  y  $b$  como constantes, se llega a la ley de Fechner:

$$s(x) = a + b \log(x + x_0)$$

En un famoso artículo, "To honor Fechner and repeal his Law" publicado en 1961, Stevens rechazaba la fórmula logarítmica propuesta por Fechner en favor de una fórmula exponencial válida, según sostenía, para todo tipo de sensaciones, incluida la visual. Stevens llevó a cabo un gran número de experimentos que sintetizó en lo que se denomina la ley de potencias de Stevens. Esta ley, en su forma general, tendría esta forma

$$\psi(I) = kI^p$$

en donde  $I$  es la magnitud del estímulo,  $\psi(I)$  una función psicofísica que relaciona la sensación subjetiva con el estímulo,  $k$  es una constante que depende del tipo de simulación y unidades utilizadas, y  $p$  es un exponente que depende del tipo de simulación y que en el caso del brillo sería del orden de 0,33 (y del orden de 0,67 para el sonido, de 1,4 para los sabores, de 0,6 para los olores...; la relación completa puede verse en el artículo citado y en otras obras de Stevens).

Ha habido un gran número de correcciones posteriores a la ley de Stevens con múltiples fórmulas alternativas. Puede decirse que proporciona una aproximación interesante pero en modo alguno concluyente. En el caso de la sensación de brillo, que es la que nos interesa, hay variantes proporcionadas por otros científicos, como Bodman, Haubner y Marsden, en trabajos publicados en 1980, que dan la fórmula siguiente:  $B = aL^p - B_0$ , donde  $B$  sería el brillo percibido,  $L$  la luminancia del estímulo,  $p$  un exponente determinado experimentalmente,  $a$  una constante que dependería de las condiciones, unidades, etc., como en la fórmula de Stevens, y  $B_0$  un factor de corrección que dependería de la respuesta neurológica al entorno, un fenómeno denominado "inducción" que lleva a alterar ligeramente la sensación percibida por la influencia del fondo.

Las dificultades de construir una formulación adecuada no se han comprendido correctamente durante mucho tiempo. Un notable investigador como Chevreul (1839) daba una recomendación, aparentemente correcta, sobre cómo construir una escala de valores, a base de superponer capas iguales de gris. Esta recomendación hace sospechar que Chevreul dió una norma que le parecía evidente sin molestarse en comprobarla. Pues la escala que así se obtiene no da la sensación, a ningún observador "normal", de estar basada en intervalos equivalentes. Dicho de otro modo, la ley de Weber-Fechner prueba que tal progresión no puede ser aritmética sino geométrica. E incluso esto último es dudoso como también hemos visto; pues entre otras

cosas no es claramente aplicable a los extremos de la escala.

La construcción de una escala de grises, de modo que resulte satisfactoria para un ojo educado en la apreciación de intervalos lumínicos, se basa, por consiguiente, en la aplicación de medias geométricas más factores imponderables que dependen, en última instancia, del ajuste de los soportes materiales concretos que se estén utilizando. Es decir, que para obtener un gris medio entre un gris claro, de 80 % de reflectancia, y un gris oscuro, de 5 % por ciento de reflectancia, deberemos crear un gris de un 20 % de reflectancia ( $5 \times 4 = 20$ ;  $20 \times 4 = 80$ ) ajustando en cada caso el soporte material para que se aproxime a esta cifra. Por otro lado, hay una aceptación general de que lo que denominamos "gris medio" corresponde a un gris que reflejaría aproximadamente el 18 % de la luz que recibe (aunque en otras referencias importantes es algo más: en el sistema de Munsell es aproximadamente 20 %). Esta aceptación general se ha traducido desde hace muchos años en el hecho trivial de que podemos ad-

quirir un gris de estas características en cualquier buena tienda de fotografía pidiendo "un gris medio".

Como la construcción de escalas es un procedimiento propio de los artefactos con que intentamos simular los escenarios reales, lo continuaremos desarrollando en el capítulo siguiente. Aquí sólo interesa introducir este tema relacionándolo con la capacidad de discriminación natural.

Además de los trabajos de Stevens, ha habido varios estudios experimentales que han buscado establecer valores adecuados para las escalas perceptuales de intensidades y, a partir de ahí, buscar alguna fórmula que se ajuste adecuadamente a los datos. La obra de Wyszecki y Stiles (p 496 y 823 de la edición citada en referencias) recoge los estudios principales que se muestran en la figura 1.52. Doy las fórmulas correspondientes a los diferentes gráficos, más que nada para hacer ver hasta qué punto no hay una respuesta clara y definitiva a la formación de escalas de valores.

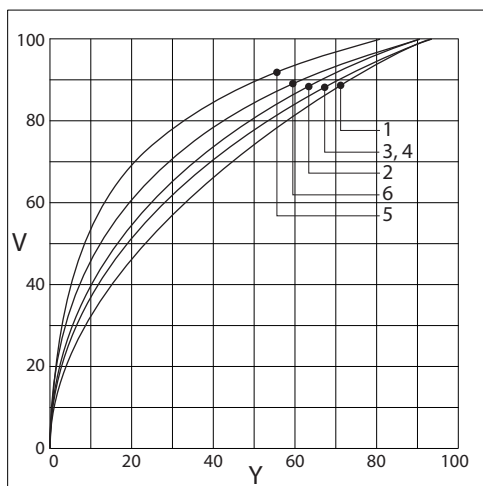


Figura 1.52 Curvas de relación entre valores de luminancia (Y) y valores de brillo o luminosidad (V) según diferentes investigadores. Las fórmulas, el contexto y las aplicaciones de estas fórmulas se indican en el texto. Adaptado de Wyszecki y Stiles (1982).

1. Priest, Gibson y McNichols (1920). Se ha utilizado en relación con el sistema de Munsell y resulta más adecuada con un fondo blanco. La fórmula es:  $V = 10Y^{1/2}$
2. Munsell, Sloan y Godlove (1933). Es una versión modificada de la anterior que funciona mejor con un fondo gris medio (de factor de luminancia 19,1). La fórmula utilizada es  $V = (1,474Y - 0,00474Y^2)Y^{1/2}$
3. Newhall, Nickerson y Judd (1943). También se ha utilizado con el sistema de Munsell y un fondo gris medio (de factor de luminancia 20). El valor  $Y_{\text{mgO}}$  con que se pondera la luminancia en esta fórmula es el correspondiente al óxido de magnesio. La fórmula utilizada es particularmente compleja:

$$100Y/Y_{\text{mgO}} = 1,2219V - 0,23111V^2 + 0,23951V^3 - 0,021009V^4 + 0,0008404V^5$$

4. CIE (1976). Es la denominada *lightness function* de la CIE en la que  $L = C$ , y en donde el factor  $Y_0$  corresponde a un blanco nominal que corrientemente se hace igual a 100. La fórmula es una raíz cúbica modificada,  $V = 25(100Y/Y_0)^{1/3} - 16$ . Hay



otras versiones posteriores de esta misma fórmula en las cuales se obtiene el valor  $L^*$  (pronunciado “el-star”), que en principio sería igual a  $V$ , mediante una fórmula algo diferente. En cualquier caso, el valor  $L^*$  (*lightness*) abarca un rango de 0 a 100 y una unidad dentro de este rango se toma como el umbral mínimo perceptible. Esto implica un rango dinámico de 100:1 que es aceptable en situaciones más o menos corrientes pero es más bien bajo, como ya hemos visto.

5. Foss, Nickerson y Granville (1944). Se ha utilizado con un sistema de codificación de colores bastante extendido, el *Color Harmony Manual*, y se basa en la ley de Weber. El negro,  $V_0 = 0$ , corresponde a “un buen pigmento negro”, con  $Y = 0,009$  y el blanco a “un buen pigmento blanco”, con  $Y = 0,0891$ . Funciona mejor con un fondo gris medio. La fórmula es  $V = 0,25 + 5\log_{10} Y$ .
6. Richter (1953). Se ha utilizado para elaborar la carta de colores alemana DIN y está basada en una versión modificada de una fórmula de Delboeuff. Funciona mejor con un fondo gris medio de luminancia aproximada  $Y = 50 \%$ . La fórmula es  $V = 6,1723 \log_{10} (40,7Y + 1)$ .

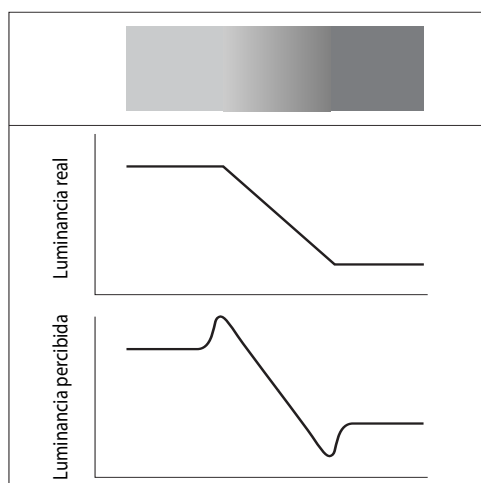


Figura 1.53 Bandas de Mach.

## Las bandas de Mach

Como decía anteriormente, el número de grises que un observador humano puede distinguir, si los valores están yuxtapuestos, es muy superior al número de grises que puede identificar en una escena. Esto es así pues, además de que la capacidad de discriminación de diferencias es muy superior a la capacidad de reconocimiento de valores amorfos, esto se relaciona con un fenómeno que permite establecer, no el número de grises que un observador medio es capaz de percibir conscientemente en una escena, sino el número de grises necesario para que una banda continua aparezca como tal sin que se produzcan efectos de “contorneado”.

Esto se relaciona con un fenómeno visual detectado por Ernest Mach en 1865 y que se conoce como “bandas de Mach”. El fenómeno puede ser analizado con instrumentos de precisión y muestra que nuestro sistema visual exagera el cambio de intensidad real que se da cuando hay un cambio relativamente abrupto en un fondo continuo. La explicación del fenómeno no es completa pero, en general, se acepta que tiene lugar a nivel neurológico y que depende de receptores que se disparan por encima de un cierto umbral mientras otros se inhiben. La figura 1.53 ilustra este efecto.

La capacidad de discriminación, teniendo en cuenta este fenómeno, se sitúa aproximadamente, según diversos experimentos, en torno al 0,5 % de la luminancia global percibida. Esto permite establecer la cifra de unos 200, como número máximo de niveles de gris que se detectan, virtualmente, en condiciones óptimas, en el caso particular de los degradados continuos y, por consiguiente, como número de grises mínimo que un sistema reproductor debería ser capaz de generar para evitar efectos de contorneado tales como los descritos. En la práctica, este valor es mucho menor y varía considerablemente según las condiciones de la escena y las condiciones de observación. Cuando se trabaja con bits, hay que escoger entre 128 (7 bpp), que sería



suficiente para la gran mayoría de los casos, y 256 (8 bpp). Se escoge este último valor por las razones dadas pero también por razones técnicas, pues resulta más conveniente empaquetar la información en paquetes de 8 bits.

## El sistema de zonas

Desde un punto de vista conceptual, pero también práctico, no es tan importante el problema técnico de construir escalas de grises, problema al que también volveremos en el capítulo siguiente, como la necesidad de contar con una guía clara e intuitiva que nos permita juzgar adecuadamente la distribución de valores de una escena.

No es nada extraño que las propuestas más interesantes en este sentido hayan venido del mundo de la fotografía.

En fotografía, la mayor o menor cantidad de luz que entra en la cámara se controla por pasos que hacen que esta cantidad sea mitad o doble que en el paso anterior. Al abrir o cerrar un punto el diafragma o al aumentar o disminuir un punto la velocidad de cierre del

obturador, el efecto es el mismo: la exposición, es decir, la cantidad de luz se duplica o se divide por la mitad. Para controlar que esto sea así, estos pasos siguen una progresión geométrica de razón raíz de 2 (los números están simplificados pero  $2 \times 2 = 4$ ;  $2,82... \times 2,82... = 8$ ). Los valores de apertura, dados en números  $f$  (igual, como ya hemos visto, a distancia focal/diámetro de apertura) son:

... 1 - 1,4 - 2 - 2,8 - 4 - 5,6 - 8 - 11 - 16...

La serie de tiempos de obturación sigue una progresión similar (los números expresan segundos):

... 4, 2, 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$ ,  $\frac{1}{15}$ ,  $\frac{1}{30}$ ,  $\frac{1}{60}$ ,  $\frac{1}{125}$

Y la serie de sensibilidades de la película, o “velocidad de la película”, sigue una progresión similar:

... 25, 50, 100, 200, 400, etc.)

La cantidad de luz que entra en una cámara con cualquiera de estos controles, sea aumentando la apertura del diafragma, sea reduciendo el tiempo de obturación (dejaremos

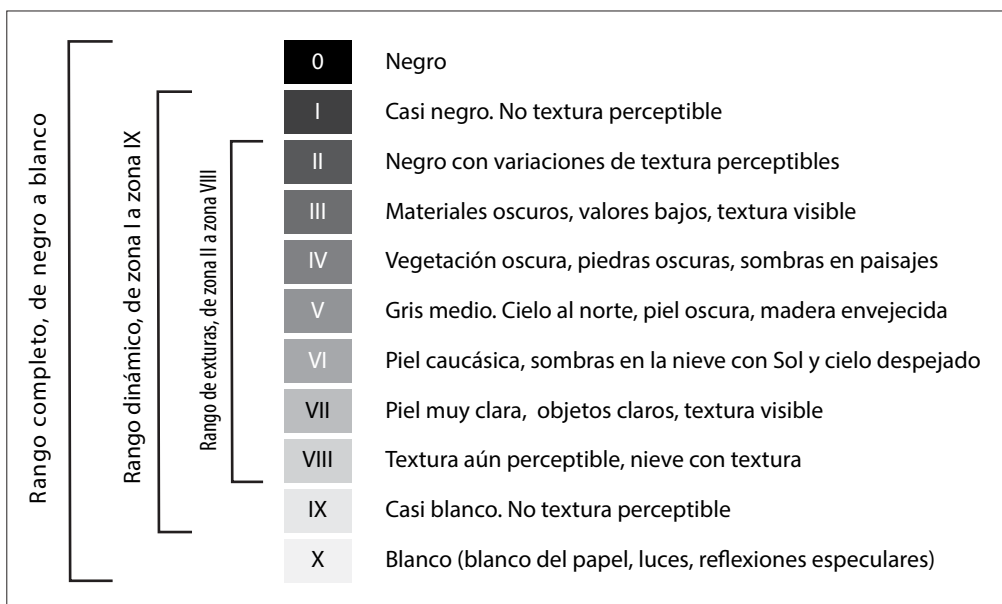


Figura 1.54 El sistema de zonas.



a un lado la sensibilidad de los sensores para simplificar), se unifica mediante lo que se denomina EV (*exposure value*, valor de exposición) que expresa diferentes combinaciones de apertura y velocidad. Así, para ISO 100, un EV = 0 corresponde a  $f/1$  y 1 s. Un EV = 1 corresponde a  $f/1$  y  $\frac{1}{2}$  s o bien a  $f/1.4$  y 1 s, etc.

Los experimentos dirigidos a la formación de escalas utilizan por lo general 11 muestras, con 0 y 10 como valores extremos, 5 como valor medio y 1,2,3,4 — 6,7,8,9 como valores intermedios. Este criterio es el mismo con el que se configura el sistema de zonas (*zone system*), un sistema que sigue siendo utilizado por los fotógrafos profesionales y que fue desarrollado y utilizado por el gran fotógrafo americano Ansel Adams (1902-1984) en colaboración con Fred Archer, hacia 1940 (basado probablemente en ideas de Denman Ross, hacia 1907) para controlar la exposición correcta de las fotografías impresas. Aunque el método de Adams, que fue utilizado también por otros grandes fotógrafos, como Edward Weston, estaba destinado a fotografías analógicas en blanco y negro, se ha mantenido como una referencia teórica importante debido a su claridad conceptual y muchos profesionales lo siguen utilizando en fotografía digital o, de modo más general, como método de análisis de escenas, fotografías e imágenes sintéticas.

Dado que este tema está relacionado con la fotografía y, por tanto, con la creación de escenarios virtuales, debería ser tratado en el siguiente capítulo. Sin embargo, el sistema de zonas es realmente un sistema conceptual que apunta precisamente a un tema de gran importancia que es la capacidad de valorar y clasificar, intuitivamente, los valores de una escena. Por esto se trata aquí en primer lugar.

El sistema de zonas se basa en el establecimiento de 11 valores de luminosidad (9 en la práctica) que sirvan de referencia para juzgar la luminosidad de las diferentes zonas de una escena real y mantener esta relación en la impresión fotográfica. Este marco conceptual es muy importante pues las mediciones de la imagen real no pueden mantenerse

en la imagen impresa debido a todo tipo de factores que afectan al proceso de revelado (en el caso de fotografías analógicas) o a los procesos de captación y manipulación posterior (en el caso de fotografías digitales), o a errores que puedan interferir en el proceso.

Las zonas se etiquetan con números romanos. Hay tres rangos de zonas: a) el rango completo, de blanco a negro, de la zona 0 a la X; b) el rango dinámico, que incluye algún tipo de información, de la zona I a la IX; c) el rango de texturas, que puede incluir información sobre la textura de los objetos, de la zona II a la VIII. Estos rangos se representan aproximadamente en la figura 1.54.

En una fotografía de una escena, cada zona representa el doble o la mitad de luminancia de la zona contigua y cada objeto refleja una cantidad de luz que se corresponde con esta luminancia captada por la cámara. Debe tenerse en cuenta que esto no se corresponde con el rango real de la escena sino con los pasos de valor de exposición de la cámara. Cada zona se corresponde de hecho con un *f-stop* (o con la mitad de tiempo de exposición). Los 8 pasos que se dan entre las 9 zonas principales representan, así, un rango de luminancias de 256 ( $2^8$ ), menos de lo que una cámara puede captar en un día soleado.

En la fotografía impresa, la zona 1 es negro “puro” (en torno a un 2 % de reflectancia real), la zona 9 corresponde al blanco “puro” (el blanco del papel que puede tener una reflectancia de un 90 % si es de buena calidad) y la zona 5 es un gris medio que se corresponde con una reflectancia normalizada de un 18 %. La posición del gris medio, el punto de equilibrio, la decide el fotógrafo basándose en su buen juicio o en el tipo de imagen que quiere obtener.

→ 2



## Escenarios virtuales

### 2.1 La luz, los objetos y el ojo en un entorno virtual

En los escenarios virtuales, como en los reales, intervienen tres factores principales: la luz, los objetos y el ojo. Para que un escenario virtual sea visible debe estar, como el real, “iluminado” por una luz virtual. Y, como en el real, esto no serviría de nada si no hubiera objetos virtuales que modulasen esta luz. Por último, tampoco veríamos nada si el escenario quedase encerrado en el disco duro y no se proyectase sobre un “ojo” o una “cámara” virtual que, a su vez, tradujera esta información a puntos de colores que se activarán sobre un monitor u otro dispositivo real que podremos contemplar con nuestros ojos.

Pero la situación es más compleja, como se revela en que la última descripción ha sido un tanto retorcida. El ojo virtual queda muy separado del ojo real y requiere varios intermediarios. De hecho, el ojo virtual comienza por ser, no un ojo, sino una proyección.

Esto implica algo más que una puntualización escrupulosa. Nuestros ojos reales son activos: se adaptan a diferentes condiciones, tanto por lo que respecta a la capacidad de discriminación geométrica como a la capacidad de discriminación de intensidades, como a la propia capacidad de discriminación de objetos.

Es precisamente esta diferencia, este ver lo visto, esta situación en la que el ojo real activo recibe la información suministrada por el ojo virtual pasivo, lo que da lugar a diversas complicaciones en donde, como veremos, se enredan problemas técnicos con problemas conceptuales que no están plenamente resueltos.

Sin embargo, podemos partir de este esquema, que replica el del capítulo anterior, como guía para esta introducción general.

Al margen de estas complicaciones, que ya se irán desplegando, en un escenario virtual tiene que haber luces, objetos y ojos.

### 2.2 Luz virtual. Unidades. Calidad lumínica

Comencemos por la luz. La primera observación es que en un escenario virtual elemental no rigen las unidades básicas que hemos visto en el capítulo anterior. Una luz, situada en medio de una escena tiene, en principio, en muchos programas de simulación, una intensidad que varía “entre 0 y 1”.

¿Qué quiere decir esto? Que si su intensidad es “0”, lo que se mostrará en el monitor será un píxel “negro”, y que, si su intensidad es “1”, lo que se mostrará en el monitor será un píxel “blanco”. ¿Qué negro? El negro más negro que pueda mostrar el monitor y que incluirá una cierta cantidad de iluminación residual más los reflejos que le lleguen de la habitación. Es decir, un negro muy poco negro. ¿Qué blanco? El blanco que el monitor sea capaz de representar con toda su energía, que no es mucha, activada para excitar al máximo los tres flujos de intensidad filtrados para producir colores primarios. Es decir, un blanco muy poco luminoso. Y, si lo imprimimos, el negro será el negro más cubriente que sea capaz de generar la impresora y el blanco será el blanco del papel. De nuevo, un negro muy poco negro y un blanco muy poco luminoso.

En un escenario virtual algo menos elemental, sí que es posible replicar, nominalmente, las unidades básicas que hemos visto en el capítulo anterior. Podemos asignar a una luz una intensidad de 10.000 cd, o de 20.000 lúmenes, o de 30.000 luxes. Pero esto es solo en teoría. En la práctica estaremos





utilizando el mismo rango que valores que proporcionan el monitor o la impresora, pues estamos viendo un mundo virtual a través de un filtro real. Un filtro que solo admite un rango de luminancias limitado y muy inferior al de las luminancias del mundo real en el cual está colocado.

Esta situación puede mejorar un poco si nos acostumbramos a trabajar en condiciones adecuadas. Un monitor bien calibrado, situado en un entorno de baja iluminación y colocado de tal modo que no lleguen reflejos a la pantalla, mostrará más valores que uno mal calibrado y mal colocado. Las diferencias pueden ser notables y pueden hacer que, por ejemplo, pasemos de ver una zona completamente oscura a ver una zona oscura en la que se pueden apreciar diferentes objetos y detalles.

Es importante no perder de vista todo esto. Primero, porque gran parte de lo que seguirá tiene que ver con técnicas que, en muchos casos, afectan a detalles sutiles que pueden perderse si las condiciones de observación no están bien ajustadas. Segundo, porque una simulación que intente replicar distribuciones de iluminación reales, acabará por ser re proyectada de tal modo que, por un lado, gran parte de los valores quedarán recortados, suprimidos y, por otro lado, estos valores deberán ser redistribuidos de modos diversos, como también veremos, sea mediante correcciones globales aplicadas mediante un ajuste del factor *gamma*, sea mediante correcciones superpuestas, globales o locales, mediante técnicas de reproyección de tonos (*tone mapping*).

## Unidades

En los entornos virtuales, por tanto, recapitulando lo dicho en el apartado anterior, se usan dos tipos de unidades.

a) Unidades convencionales. El rango de referencia en este caso es 0,0 a 1,0 si bien se pueden asignar valores inferiores (negativos) o superiores (sin límite).

Estas unidades, habituales en la mayoría de los programas de simulación, no se corresponden con ningún valor físico real sino con el monitor que estemos utilizando. Como ya he dicho, “0,0” quiere decir “píxel de intensidad mínima” mientras que “1,0” quiere decir “píxel de intensidad máxima”. Y “mínimo” y “máximo” son valores que dependen del monitor y que pueden ser del orden de unas 0,1 cd/m<sup>2</sup> y 300 cd/m<sup>2</sup> en el mejor de los casos, como hemos visto en el capítulo anterior, aunque en la práctica es posible que este último valor esté en torno a las 100 cd/m<sup>2</sup>.

Cuando se usan este tipo de unidades la intensidad se controla a veces con un “multiplicador” (*multiplier*). El multiplicador puede recibir valores negativos en cuyo caso la luz restará iluminación a la zona de la escena sobre la que caiga. Esta característica, exclusiva de los mundos virtuales, nos permite utilizar un truco, que todavía se usa en algún caso, para simular sombras indirectas, atenuadas, por medio de luces suaves que crean, directamente, sombras.

Por otro lado, el multiplicador puede recibir valores superiores a 1,0. ¿Pero qué significa esto si acabamos de decir que “1,0” significa “píxel de intensidad máxima”. Si hacemos que el multiplicador sea de 10,0 el valor de un punto “blanco” seguirá siendo el mismo. ¿Entonces?

La respuesta es que esto altera la distribución de los valores en la escena. Al aumentar el valor del multiplicador estamos estirando los píxeles, con valores altos, hacia el máximo. Muchos de ellos alcanzarán este límite y se convertirán en blancos (sin poder pasar de ahí). La escena quedará “quemada”, con un exceso de valores altos.

b) Unidades fotométricas. En muchos programas de simulación se pueden utilizar, como veremos, luces “fotométricas”. Este tipo de luces tienen propiedades diferentes de las de las luces simples o “estándar”. Y utilizan las unidades físicas que hacemos servir en el mundo real para calibrar intensidades radiantes (lúmenes, candelas), intensidades



aparentes (luminancias) e intensidades irradiantes (luxes). Esto nos permite componer escenas en las que la *relación* de valores luminosos sea semejante a una escena real. Pero hay que insistir en que los valores que percibimos siguen estando restringidos, y a menudo recortados y alterados, al mismo rango que el de las unidades arbitrarias que dependen de las características del medio, monitor o impresora, a través de cual los vemos.

### La calidad lumínica

En el capítulo anterior hemos visto como la calidad lumínica depende, además de aspectos imponderables, de la temperatura de color y de ciertos rasgos generales relacionados principalmente con el contraste global y la dirección de las luces principales y secundarias.

Por lo que respecta a lo primero, la primera observación será, de nuevo, que para conseguir una tonalidad correcta, un balance de blancos bien equilibrado, antes de nada hay que asegurarse de que el monitor está correctamente calibrado y que las condiciones de trabajo son también correctas. A partir de ahí, es posible introducir pequeñas variaciones en la tonalidad de la escena variando ligeramente el color de las luces principales y secundarias. Según los tipos de luces utilizados y según el sistema de cálculo, varían las posibilidades, que se irán viendo en los capítulos correspondientes. También cabe hacer ajustes globales a toda la escena a partir de los controladores del control de exposición, otro tema al que también volveremos, y mediante técnicas de postproducción.

Por lo que respecta a lo segundo, las observaciones generales que se han dado en ese capítulo son igualmente aplicables a los escenarios virtuales. La iluminación de una escena real, sea buscando una hora del día y un punto de vista que nos muestre la naturaleza con ciertas características que nos interesa captar, sea disponiendo, como en un estudio, de luces principales y secundarias que realcen ciertos aspectos y aminoren

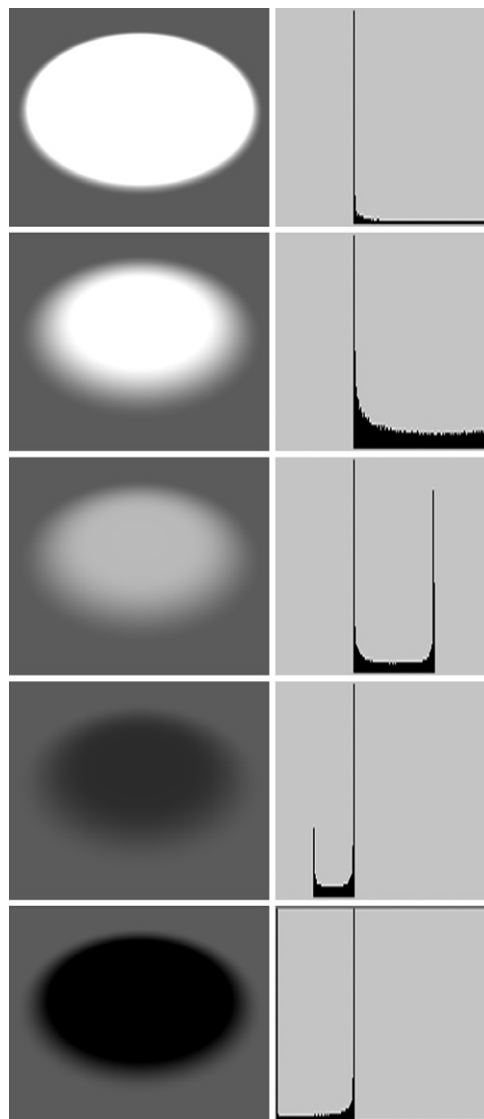


Figura 2.1 Desplazamiento del histograma con luces simples al variar los valores del multiplicador. Las imágenes de la izquierda muestran el efecto sobre un plano, de una luz focal con una zona central uniforme (hotspot), de 20° y una zona total (falloff) de 45°. Las imágenes de la derecha muestran el histograma correspondientes. Valores del multiplicador: a) +10,0, b) +1,0 (predeterminado), c) +0,5, d) -0,25, e) -1,0



otros, es similar a la de una escena virtual. También, como veremos, para preparar una escena virtual deberemos razonar como un fotógrafo y ponderar adecuadamente si nos interesa realzar el contraste, con zonas muy claras o muy oscuras o bien si nos interesa crear transiciones suaves y sutiles.

La diferencia es práctica pues, por un lado, costará mucho más trabajo conseguir las calidades propias de los escenarios reales, pero, por otro lado, costará mucho menos trabajo montar luces virtuales en cualquier posición. Y, por añadidura, tenemos a nuestra disposición no solo medios de ajustar con facilidad las diferentes intensidades de las luces sino, también, de ajustar con facilidad el contraste de salida.

## 2.3 Tipos de luces

### Luces virtuales simples

En todos los sistemas de simulación visual hay un sistema básico de simulación de la iluminación que se basa en un algoritmo muy simple. Consiste en computar el producto vectorial unitario de los vectores correspondientes a la luz y a la normal a la superficie. En muchos de estos programas se denominan, a estas luces, luces “estándar” cuando se utilizan otros tipos de luces, para diferenciarlas. Y las luces “no estándar” se denominan, en muchos casos, “fotométricas”.

**Tipos.** En realidad, puede decirse que hay un único tipo principal de luz virtual simple o básica: una luz puntual que tiene una posición determinada en el espacio y que irradia en todas direcciones. Los otros tipos pueden considerarse variantes de este tipo básico. Atendiendo, por tanto, a su uso corriente puede decirse que hay tres tipos principales de luces estándar que se crean con facilidad, con técnicas muy elementales: luces puntuales (*point lights*) u omnidireccionales (*omni lights*), focales (*spot lights*) y direccionales (*directional lights*). Las luces *puntuales* tienen

una posición en el espacio, desde donde irradian en todas direcciones una determinada intensidad (predeterminada a 1,0) y un color (predeterminado a blanco). Las luces *focales* son similares a las puntuales (sus propiedades básicas son exactamente las mismas) y tienen también una posición dada, pero irradian en una dirección principal que se controla por medio de un cono interno y un cono externo que determinan cómo se distribuye su iluminación. En la zona que abarca el cono interno la intensidad es máxima. En la zona comprendida entre el cono interno y externo la intensidad pasa de ser máxima a ser nula. Las luces *direccionales* no requieren una posición en el espacio e irradian en una dirección única. En algunos programas no se les asigna una posición sino tan solo una dirección, mientras que en otros se les asigna una posición en la escena para poder controlarlas mejor, lo que implica que también pueden contar con un cilindro interno y un cilindro externo que cumplen la misma función que la focal, aunque el modo en que iluminan a los objetos es independiente de su posición.

Para crear estas luces se utilizan funciones que piden los valores específicos que hemos citado en cada caso: posición, intensidad, color, y, en el caso de las dos variantes citadas, dirección, ángulo de los conos internos y externos o amplitud del cilindro de la zona activa.

Las luces estándar se basan en un algoritmo que está incorporado a todas las tarjetas gráficas que utilizan Direct3D u OpenGL, lo que hace que su velocidad de procesamiento sea aún mayor. Este algoritmo se da en dos variantes igualmente estandarizadas: *flat shading* (sombreado plano) y *smooth shading* o *Gouraud shading* (sombreado suavizado).

Con la primera variante se computa el producto vectorial punto del vector Luz, normalizado e invertido, con el producto vectorial punto del vector Normal a la superficie, igualmente normalizado. Como este producto es el coseno del ángulo formado por los dos vectores y este coseno es 0 cuando el ángulo es 90°, y 1 cuando el ángulo es 0°, resulta



sencillo ajustar la intensidad en función de un valor lineal comprendido entre 0 y 1, lo que se corresponde exactamente con la menor o mayor incidencia de la luz sobre la superficie. Este procedimiento fue introducido por W. J. Bouknight en 1970.

Con la segunda variante se añade al cálculo anterior una interpolación que se procesa en otro estadio de la *rendering pipeline* (véase el capítulo 4). Los valores de iluminación se incorporan a los vértices, como un atributo que se guarda junto con las coordenadas geométricas de cada vértice. Esta información, propia de un algoritmo de procesamiento denominado *vertex shader* y que se describirá con algo más de detalle en el capítulo 4, se pasa a un estadio posterior a cargo principalmente de otro algoritmo principal denominado *píxel shader* en donde los valores de vértices adyacentes se interpolan y los valores resultantes se asignan a los píxeles de salida. De este modo se simula una curvatura que realmente no existe, pues la geometría de base sigue siendo la misma como puede apreciarse en los bordes de una esfera que seguirán siendo fragmentos rectos. El proceso de interpolación se denomina “sombreado suavizado” o “sombreado Gouraud” (*Gouraud shading*), por el nombre del investigador Henri Gouraud que lo introdujo en 1971. Variantes mejoradas de este algoritmo están incorporadas a propiedades de materiales, como los *shaders* de Phong (1975) o Blinn (1977) corrientes en cualquier programa de simulación. Las figuras adjuntas, 2.2 y 2.3, ilustran las descripciones anteriores.

Las luces estándar tienen una serie de limitaciones importantes que se detallarán más adelante. La principal es que no computan la interacción entre los objetos por lo que las partes no iluminadas aparecerían, en principio, completamente negras. En programas antiguos se utilizaba un cuarto tipo de luz para evitar que las sombras de los objetos resultaran negras, una luz denominada de *ambiente*, que también se ilustra en la figura 2.3 y que iluminaba a los objetos por igual en todas direcciones. Este tipo de luz está prác-

ticamente en desuso, pues, o bien se utilizan luces avanzadas con sistemas de cálculo que simulan correctamente la interacción entre objetos, o bien la mayoría de los usuarios prefiere utilizar luces secundarias para simular esta interacción.

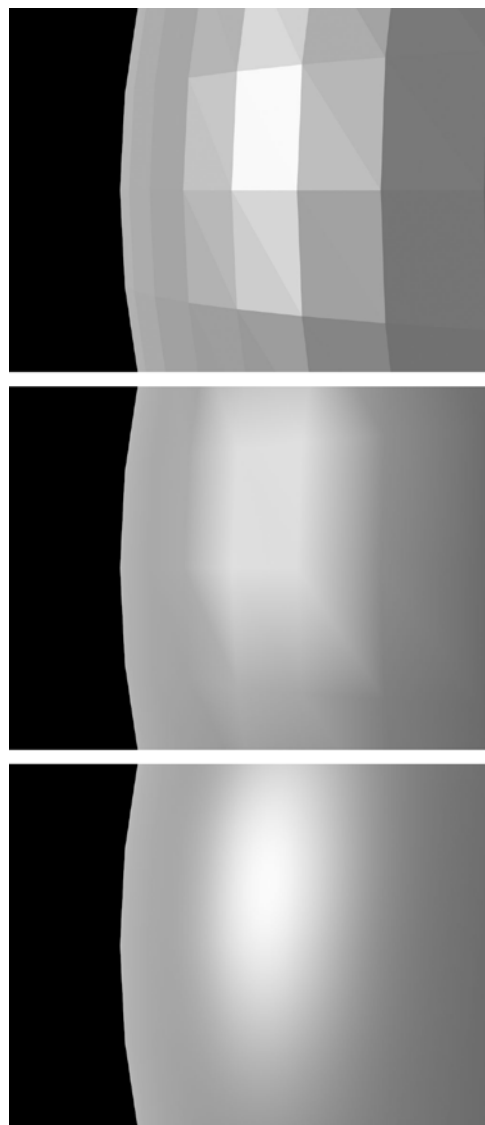


Figura 2.2 Algoritmo básico de cálculo con luces estándar. Flat shading. Gouraud shading.

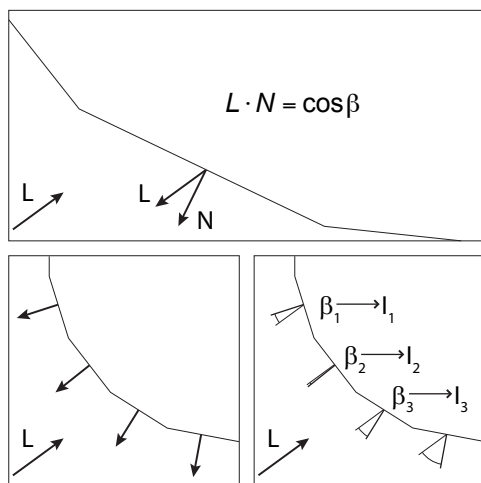


Figura 2.3 Flat shading.

**Propiedades.** Las luces estándar tienen dos propiedades principales: su intensidad y su color. La *intensidad* se computa, como ya he dicho, mediante un valor convencional que está referido a la gama de intensidades del sistema. Una luz de intensidad 1,0 es una luz que altera de un modo claramente apreciable un plano de color gris medio produciendo un degradado. La figura 2.5 muestra un ejemplo del efecto producido por una misma luz con tres intensidades de 0,5, 1,0 y 1,5. El *color* es un parámetro que se asigna de modo exactamente igual a como se asigna el color de un material, pero que tiene el efecto de teñir todos los objetos que ilumina de este color. El color predeterminado es el blanco y en la gran mayoría de los casos se deja así.

Además de estas propiedades principales hay otras que varían según los programas de aplicación y que se verán con algo más de detalle en el capítulo siguiente y en los apartados correspondientes de este, como el tipo de sombras. Como ya hemos visto, las luces directas y focales incorporan un cono o un cilindro que limita su acción al volumen dado por estas formas. Pero hay otras propiedades de interés, y una de las propiedades adicionales que se pueden encontrar en muchos programas es la simulación de la atenuación.

**Atenuación con la distancia.** Otra de las limitaciones de las luces estándar es que no incorporan la atenuación con la distancia, una de las propiedades características de las luces reales. Pero la atenuación puede incorporarse al cálculo mediante parámetros adicionales utilizados en muchos programas. La mayoría de estos parámetros están basados en fórmulas internas que utilizan la distancia entre la luz y el objeto, que puede computarse fácilmente a partir de los dos vectores de posición. Dado este parámetro adicional,  $d$ , el factor de atenuación puede computarse mediante una fórmula del siguiente tipo:

$$\text{Factor atenuación} = 1 / (K_c + K_l d + K_q d^2)$$

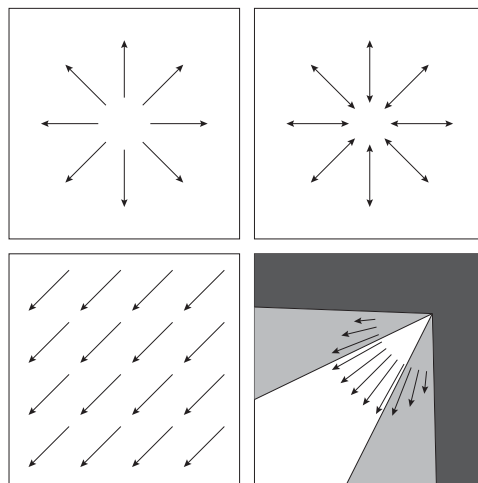


Figura 2.4 Tipos de luces estándar: Omidireccional. Ambiental. Direccional. Focal.



Figura 2.5 Plano iluminado por una luz omnidireccional, centrada sobre el plano y con intensidades nominales de 0,5 (izquierda), 1,0 (centro), 1,5 (derecha).



Donde  $K_c$  es una constante simple,  $K_l$  es una constante de atenuación lineal y  $K_q$  es una constante de atenuación cuadrática.

En el capítulo 5 veremos ejemplos de cómo pueden utilizarse estos parámetros en la práctica.

**Atenuación por oclusión parcial.** Una última limitación de las luces estándar es que no computan la atenuación por oclusión parcial. En las esquinas o, en general, cuando un objeto se encuentra con otro objeto, se produce un oscurecimiento debido a un efecto de apantallamiento, de obstrucción de la iluminación global que viene de todas direcciones. Cuando este efecto está ausente, los objetos parecen flotar de un modo irreal, tal como se muestra en la figura 2.6 en la que se han resaltado “puntos” más o menos cercanos a otro objeto para ilustrar cómo este objeto actúa como una pantalla que reduce la intensidad de la luz que viene de todas direcciones.

La atenuación por oclusión parcial puede computarse de modo independiente mediante algoritmos especiales, incorporados en algunos programas y que también se describirán en los capítulos siguientes.

**Ventajas y desventajas.** Una luz “estándar” es una luz que simula la iluminación del modo más simple posible: mediante la asignación, a un punto del espacio, de unos valores convencionales que gradúan la intensidad a partir de los valores disponibles en el monitor. Es importante tener bien presentes las ventajas y los inconvenientes de semejante simplificación. Las ventajas derivan de la simplicidad de cálculo, pues todo se reduce básicamente a multiplicar dos vectores, como hemos visto. Las desventajas son: a) se trata de “un punto”, sin extensión: esto quiere decir que, en principio, no hay penumbra, pues un punto ilumina o no ilumina y las sombras resultantes pasan, sin transición de blanco a negro, a umbra; b) la iluminación solo depende del ángulo y es independiente de la distancia, tanto da que la luz esté pegada al objeto como a varios metros de él; c) los valores de luminosidad

emanan exclusivamente de la luz, los objetos no tienen capacidad de redireccionarla: esto quiere decir que las partes no iluminadas quedarían, en principio, totalmente negras al no computarse el reflejo de otros objetos cercanos.

La diversidad de tipos y técnicas no es ni más ni menos que el resultado de intentar reducir estos inconvenientes si bien el último, el más importante, no tiene solución y requiere utilizar sistemas de iluminación avanzada.

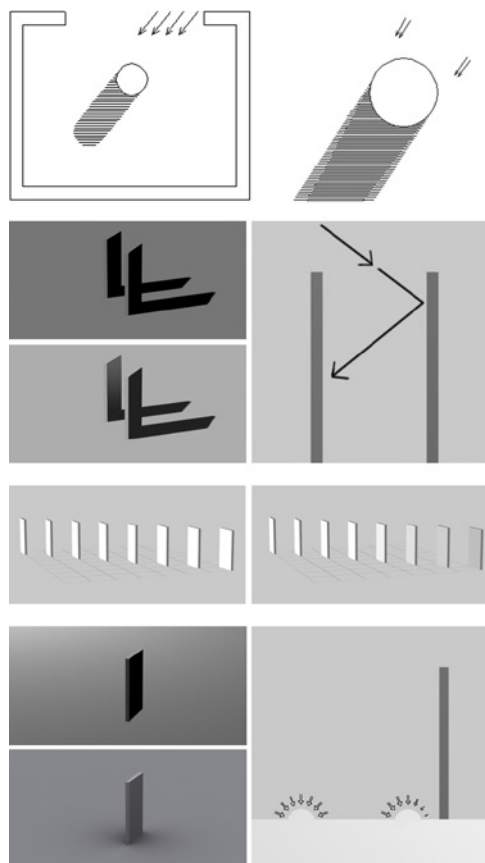


Figura 2.6 Limitaciones principales de las luces estándar: a) luz puntual sin extensión que no crea zonas de penumbra; b) no interacción entre objetos; c) no atenuación por distancia; d) no atenuación por oclusión parcial.





## Luces básicas con propiedades de área y volumen (extensas)

La primera limitación citada en el apartado anterior es que las luces estándar son puntuales. Esto es una ventaja importante en términos de rapidez y de facilidad de uso. Pero tienen el inconveniente, ya comentado, de que no se crean efectos de umbra y penumbra como ocurre con todas las luces reales.

Muchos programas incorporan recursos para simular la extensión, sea por medio de una superficie virtual sea por medio de un volumen virtual. En estos casos, la propiedad de área o volumen aparece como un parámetro adicional que se añade a la lista de parámetros corrientes de una luz estándar. Las luces extensas pueden ser de varios tipos. Las luces *de área* incluyen, por lo general, dos formas características: rectángulos y discos. También hay luces “lineales” pero suele ser más recomendable utilizar una luz de área rectangular muy alargada. Las luces *volumétricas* (incluidas a veces en un único tipo de luces “de área”) incluyen esferas y cilindros.

Los únicos inconvenientes de estas luces son, en primer lugar, que afectan a la velocidad del cálculo, debido a que deben enviarse múltiples rayos para simular la extensión en lugar de un único haz, como ocurre en las luces puntuales. Sin embargo este inconveniente se reduce cada vez más debido a la creciente potencia de ordenadores y tarjetas gráficas. Otro inconveniente, tampoco muy importante, es que requieren más ajustes pues debemos adaptar el área o volumen a cada caso particular.

Las propiedades adicionales que proporcionan las luces extensas afectan a las sombras. Véanse las figuras incluidas en el apartado sobre sombras para comprobar la diferencia entre el tipo de sombra dado por una luz puntual y una luz extensa. Y una ventaja adicional importante de las luces extensas, como también se verá en dicho apartado, es que podemos utilizar sombras de *ray trace*, más precisas y que requieren menos ajustes.

Los programas que proporcionan recursos para simular luces extensas también permiten simular la propia luz, es decir, hacer que sea visible en la escena. Pero en la gran mayoría de los casos esto no será de gran ayuda porque lo único que ocurrirá es que aparecerá una superficie o un objeto uniformemente blanco. En general, será preferible simular el objeto luminoso por medio de un objeto autoluminado. Véase el apartado sobre simulación de luminarias, más adelante.

Por último, hay que añadir que también es posible utilizar objetos que irradian luz y que crean de modo natural el efecto de una luz extensa. El problema es que la irradiación y la distribución de la luz proporcionada por este tipo de objetos no es tan efectiva, en la mayoría de los sistemas de iluminación avanzada, como la proporcionada por las luces virtuales, por lo que solo es recomendable utilizarlas como recurso complementario. Pero hay excepciones que también se verán más adelante.

## Luces fotométricas

Las luces fotométricas se utilizan en sistemas de cálculo de iluminación avanzada para obtener resultados más realistas y que permiten, por añadidura, analizar las características de iluminación de una escena determinada.

Las luces fotométricas se distinguen de las estándares por tres diferencias importantes: a) las unidades no son convencionales sino reales, y se dan en las unidades corrientes que hemos recordado en el capítulo anterior, lúmenes o candelas para la especificación de la propia fuente y luxes para la especificación a una determinada distancia; b) la atenuación está incorporada al comportamiento normal de la luz (esto es precisamente lo que permite especificar la iluminación en luxes a una determinada distancia); c) la distribución puede modificarse para simular el comportamiento de luces reales; d) se pueden configurar como luces extensas, de área o volumen.

Además de estos, y según los diversos programas, las luces fotométricas incluyen





otros parámetros adicionales que se verán en el capítulo 5 y con los ejemplos correspondientes.

La especificación de la *distribución* es importante y, si se quiere hacer una simulación precisa, puede requerir utilizar programas complementarios. En principio, una luz fotométrica incluye tres o cuatro tipos de distribuciones pero, si se cuenta con ella, la más importante es la cuarta. La primera, una distribución *isotrópica o uniformemente esférica*, indica que la luz se distribuirá por igual en todas direcciones, algo que jamás ocurrirá exactamente así en la realidad a no ser que seamos capaces de hacer que una luminaria perfectamente regular flote en el espacio. La segunda, una distribución *semi-isotrópica o hemisférica o uniformemente difusa*, indica que la luz se distribuirá por igual en una sola dirección, en torno a un hemisferio centrado en la luz. La tercera, *focal*, se basa en una distribución isotrópica pero limitada a un cono dentro del cual se puede especificar una zona central de intensidad uniforme (dada por un ángulo específico, denominado generalmente, ángulo propio del haz o *beam angle*) y una zona periférica de intensidad decreciente (dada por otro ángulo suplementario, denominado generalmente ángulo de campo o *field angle*). La cuarta, se basa en una *distribución específica de la intensidad luminosa* (a veces denominada LID, *luminous intensity distribution*, en la literatura anglosajona).

En 1984, Verbeck y Greenberg proporcionaron una descripción adecuada de fuentes de luz artificiales por medio de tres parámetros que pueden incorporarse a su simulación por medios informáticos: su geometría, su distribución espectral y la distribución de su intensidad luminosa. Los ejemplos que se proporcionaron en este artículo fueron los primeros que simulaban de un modo adecuado la distribución variable de intensidades por una fuente de luz.

La *geometría* de la fuente de luz se modela generalmente, tal como hemos visto, por medio de un punto (0D), una línea (1D) o una superficie (2D), las tres variantes que

utilizaron Verbeck y Greenberg en su artículo y que siguen siendo las más utilizadas. La *distribución espectral* puede incorporarse a la descripción de la luz si bien, en la gran mayoría de los casos, este parámetro se limita a la especificación del color de la luz en términos de temperatura de color, con el blanco normalizado D65 (6.500 °K) como referencia principal. La *distribución de intensidades* es la parte principal y se modela por métodos analíticos o experimentales.

Las características y los modos de representar estas distribuciones de intensidad, por medio de diagramas goniométricos o polares, ya se han resumido en el capítulo anterior tomando como referencia luces reales. Partiendo de la base de que ya contamos con una descripción adecuada de esta distribución, el problema con que nos encontramos será simplemente representar estos datos digitalmente del modo más adecuado.

Para especificar la distribución se utilizan archivos específicos que dan una descripción de esta distribución por medio de una “red fotométrica” (*photometric web*) en un determinado formato. El más utilizado es el formato denominado IES (*illuminating en-*

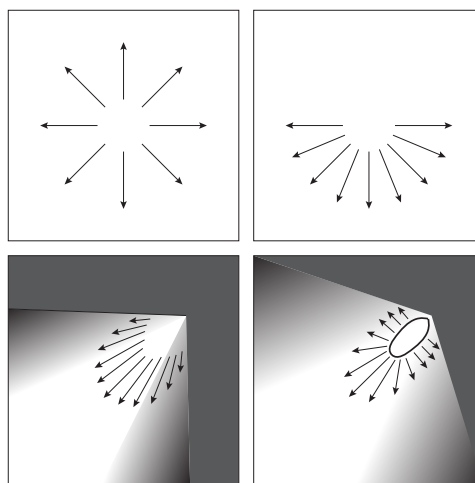


Figura 2.7 Luces fotométricas. Distribuciones características: a) esférica uniforme; b) hemisférica uniforme; c) focal; d) basada en archivos con parámetros polares.



gineering society), desarrollado en Estados Unidos por la IESNA (*Illuminating Engineering Society of North America*, <http://www.iesna.org>) y que formula la distribución de la intensidad luminosa en los nodos de una red 3D a partir de una serie de ángulos fijos en un plano horizontal y vertical de modo que se correspondan con un diagrama goniómetro (tipo C, uno de los más corrientes). Los datos están grabados en un fichero en formato ASCII con un determinado protocolo. Además del formato IES, pueden encontrarse descripciones de este tipo en otros formatos, como el formato LTLI (creado por el *Danish Illuminating Laboratory*) utilizado en los países escandinavos) o el formato CIBSE (*Chartered Institution of Building Services Engineers*, utilizado en el Reino Unido). Otro formato de referencia importante es EULUMDAT, un formato europeo (<http://www.helios32.com/Eulumdat.htm>).

Un archivo IES tiene una estructura en la que se incluyen una serie de datos sobre una determinada lámpara (fabricante, modelo, características generales, etc.) y una serie de líneas en que se dan ángulos y valores de intensidad correspondientes a esos ángulos. Para que la distribución de estos valores resulte adecuada, es necesario utilizar un programa que permita modelar de modo continuo la distribución y, a partir de ahí, extraer los datos discretos para grabarlos en formato IES. O bien contar con una buena colección de diagramas comerciales que se pueden obtener por internet. En el capítulo 5 se dará más información sobre este tipo de archivos y se proporcionarán tres archivos genéricos, generados con un programa especial, que pueden replicarse con facilidad si al lector le interesa hacerlo.

### **Luz solar. Luz celeste. Sistemas de luz diurna**

*Luz solar.* La luz solar se puede simular por medio de una luz directa, que puede ser una simple luz estándar. Una luz estándar puede cumplir adecuadamente este cometido pues

dos de sus limitaciones características, el ser puntual y el no computar la atenuación, pueden, hasta cierto punto, pasarse por alto.

Por un lado, el tamaño del Sol se proyecta sobre nuestra pupila con una extensión relativamente pequeña, del orden de los 32' de grado, como hemos visto en el capítulo anterior, por lo que, hasta cierto punto, y según el tipo de aplicaciones, puede considerarse como puntual. Y por otro lado, aunque el Sol no está en el infinito sus rayos son prácticamente paralelos y la distancia es tan grande que los efectos de atenuación pueden pasarse por alto.

Así que puede utilizarse, en principio, una luz estándar directa, de rayos paralelos y sin atenuación. Sin embargo, hay dos aspectos importantes que justifican utilizar otro tipo de luces si queremos obtener una simulación más precisa.

El primero es la *intensidad* y el *rango dinámico*. Asignar al Sol un valor convencional de 1,0 quiere decir, simplemente, asignarle el valor máximo dentro de un rango de valores que en principio va de 0 a 255. Esto queda demasiado lejos de los valores que hemos resumido en el capítulo anterior y daría como resultado una escena muy plana. Si aumentamos la intensidad, con el multiplicador, todo lo que conseguiremos, como ya he dicho, es desplazar los valores con el resultado de que todos los objetos claros aparecerán quemados, blanqueados. Por otro lado, si queremos que la escena simule una escena real necesitamos utilizar valores reales que nos permitan, en principio, asignar a los objetos reflectancias adecuadas y a la propia atmósfera valores proporcionales a la relación real entre Sol y atmósfera.

El segundo es la simplificación que supone tratar al Sol como una luz puntual en lugar de como una *luz extensa*. Aunque esto pueda resultar aceptable en muchos casos no lo será en casos más exigentes. El resultado de esta simplificación es que las sombras aparecerán iguales, tanto si están cerca del objeto que las produce como si están muy lejos. Pero esto no es así como sabe cualquiera que se haya



paseado por un exterior fijándose un poco en las sombras. Como ya he comentado en el capítulo anterior, cuando las sombras arrancan de un objeto son nítidas y a medida que se alejan se hacen más difusas. Para conseguir este resultado necesitaríamos simular el Sol por medio de una luz extensa, lo que resulta algo engorroso en la práctica. O, mejor, utilizar un programa que permita simular directamente este efecto como veremos en los ejemplos del capítulo 5.

**Luz celeste.** Una luz cenital (*skylight*, *sky dome*, *environmental sphere*) es un tipo de luz especial que rodea la escena y la ilumina desde todas las direcciones. Puede utilizarse como una fuente de luz homogénea o asignarle un mapa que incorpore variaciones en la distribución. En este caso, la intensidad de la luz varía en función de la imagen proyectada. Si la imagen proyectada es una imagen en formato HDR, la calidad de iluminación obtenida puede ser muy superior a la que se consigue por otros medios tal como también veremos en los capítulos que siguen y que

tratan de modo específico de cómo utilizar este tipo de archivos para los métodos que se denominan genéricamente IBL (*image based lighting*). Aquí solo se discutirán las cuestiones de fondo, es decir, qué recursos digitales hay disponibles para simular esta iluminación y cómo se relacionan con la luz cenital real cuyas características principales se han discutido en el capítulo 1.

En el capítulo anterior también hemos visto que se utilizan diferentes modelos para representar la irradiación celeste. Puede decirse, recapitulando, que contamos con cuatro referencias o modelos principales: a) cielo uniforme; b) cielo no uniforme cubierto; c) cielo no uniforme despejado; d) cielo parcialmente cubierto. Se trata ahora de ver, en general, cómo podrían incorporarse estos modelos a programas de simulación.

El primer modelo (a) no presenta ninguna dificultad técnica. En todos los programas de simulación hay un parámetro que asigna un color y una intensidad al “cielo” (el color de fondo de la escena). Al activar el cálculo,

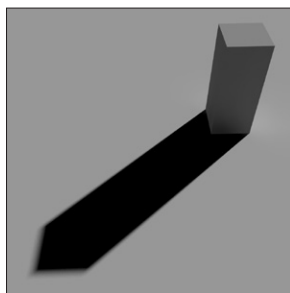


Figura 2.8 Luz solar.

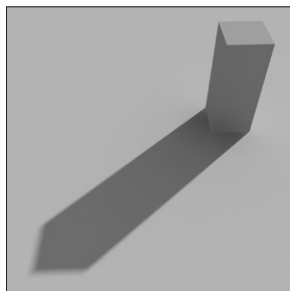
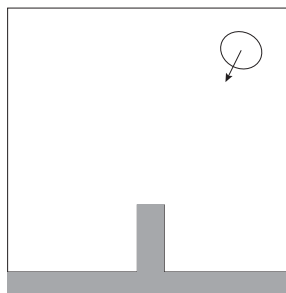
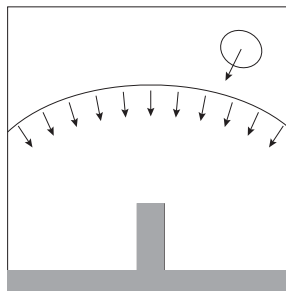


Figura 2.9 Luz solar y luz celeste.





cualquier rayo enviado desde la cámara para computar un valor se encontrará o bien con una superficie o bien, directa o indirectamente (en el caso de sistemas de iluminación avanzada que pueden computar el rebote de los rayos) con un “fondo”. Este fondo no es sino una asignación numérica similar a la asignación de color de un material. Y si el cielo es uniforme será la misma para cualquier dirección.

El segundo modelo (*b*) puede simularse por medio de mapas procedurales que asignan variaciones a los fondos. En el capítulo 5 veremos algunas de las posibilidades con técnicas y programas de aplicación concretos.

El tercer modelo (*c*) es similar al anterior. También es posible simular una variación regular por medio de algoritmos procedurales que introduzcan variaciones e incluso relacionarlas con modelos concretos de distribuciones.

En estos dos últimos casos se pueden utilizar modelos, como los de la CIE o Perez All Weather, relacionados con datos estadísticos de regiones geográficas y de fechas y horas determinadas.

El cuarto modelo (*d*) es más complejo pues “parcialmente cubierto” quiere decir que hay nubes con formas particulares. Pueden utilizarse datos estadísticos como en los casos anteriores. Pero cabe también la posibilidad, más interesante, de utilizar un mapa ligado a un archivo radiante, en formato HDR, que simule la distribución de luminancias de un cielo real.

Es decir, podemos generar un archivo radiante a partir de una serie de fotografías, con los procedimientos que se verán en el capítulo 4, y asignar este mapa a una esfera virtual que rodee la escena. El resultado será una distribución más cercana a lo que ocurre en casos reales.

En este último caso el problema que podría presentarse es el opuesto a los anteriores: la simulación estará basada en una información excesivamente particular; por ejemplo, un cielo determinado, con unas nubes determinadas, situadas a lo largo de unas montañas determinadas a una hora determinada.

Pero las diferencias pueden resultar imperceptibles y, en la práctica, según los casos, se puede acabar utilizando cualquier mapa “interesante” que proporcione variaciones en la iluminación ambiental similares a las que se darían en la escena que se quiere simular. Todo dependerá del caso a que se esté aplicando la simulación y de hasta qué punto se requiere que esta sea fiel a las circunstancias concretas del caso.

### **Sistemas de luz diurna (integración de luz solar y luz celeste)**

La mayoría de los programas informáticos proporcionan un recurso denominado “sistema de luz diurna” o “solaroscopio” o “iluminación de entorno”, etc. Estos sistemas integran, de diversos modos, una luz solar y una luz celeste. Se trata, por tanto, de un simple recurso técnico que agrupa a los dos anteriores (el Sol y la luz celeste) y también se detallará en los ejemplos dados en el capítulo 5.

### **Simulación de luminarias**

Una de las muchas cosas raras que hay que hacer al crear un escenario virtual es que, en la mayoría de los casos, se requieren dos objetos para simular una luz: un objeto que envíe rayos de luz a la escena pero que es invisible, pues a menudo no es más que un punto sin dimensión o un vector que señala una dirección y, por otro lado, un objeto que no envíe rayos pero que aparezca como un objeto luminoso.

Hay varios métodos principales para conseguir este efecto.

El más corriente es crear un objeto al que se ha aplicado un material autoiluminante. Un material autoiluminante muestra el mismo color en todas direcciones, con independencia de la luz que incida sobre él. Si este color es blanco o casi blanco, el efecto es que el objeto irradia luz. Se pueden introducir variaciones en este efecto: con una gradación uniforme o variable según la orientación o por medio de una textura autoiluminante. Véanse los ejem-



plos del capítulo 5 para comprobar paso a paso cómo pueden aplicarse estos efectos.

Otra alternativa es situar la luz dentro del objeto luminaria y asignar a este objeto un material traslúcido. El material traslúcido deja ver la luz que está tras él y la dispersa de un modo característico. Esto requiere el uso de materiales especiales por lo que me remito al libro de simulación de materiales, en donde también se trata este tema.

## 2.4 Luces y sombras. Introducción a los tipos de sombras y sistemas de cálculo

A diferencia de lo que ocurre en los escenarios reales, en los virtuales las luces y las sombras siguen procesos independientes. Esto quiere decir que un determinado tipo de luz puede escoger entre varios sistemas de cálculo de sombras. Y esto puede resultar bastante desconcertante para los que comienzan a utilizar un programa de simulación, pues pueden encontrarse con hasta media docena de posibilidades. Afortunadamente, en la teoría estos sistemas se reducen a tres principales; en la práctica, con un programa corriente se reducen a dos y, si se utilizan luces extensas, se reducen a uno.

Los tres sistemas de cálculo principales son volumen de sombra (*shadow volume*), mapa de sombra (*shadow map*) y trazado de rayos (*ray trace*). Los más corrientes en programas de simulación dirigidos a crear imágenes estáticas son los dos últimos. Y el más recomendable, si se utilizan luces extensas y el tiempo de procesamiento no es problemático, el tercero.

### Volumen de sombra (*shadow volume*)

Es una técnica presentada por primera vez por Frank Crow en 1977 (véase la referencia) y que fue ampliada posteriormente por Tim Heidmann, en 1991, por medio de *stencil buffers* (ver la descripción de la *rendering pipeline* que se da en el capítulo 4) y por otros autores

posteriores que introdujeron refinamientos en la técnica (Kilgard, 1999; Carmack, 2000; Evert y Kilgard, 2002). La incorporación de *stencil buffers* a todas las tarjetas gráficas avanzadas ha recuperado esta técnica para su uso en juegos y aplicaciones en tiempo real.

La idea básica, que se ilustra en la figura 2.10 por medio de un esquema que presenta una sección del volumen, es crear para cada objeto que se encuentre en el camino de la luz, un volumen que queda definido por la propia luz y los planos tangentes al objeto visibles desde la luz (en la figura se muestran los cuatro principales). Cada punto que queda dentro del volumen está en la sombra y cada punto fuera del volumen queda iluminado. Esto permite calcular con relativa facilidad las sombras de todos los objetos de la escena. Tiene el inconveniente de que no funciona bien con objetos que incluyen superficies curvas o que arrojan curvas sobre sí mismos.

Aunque una de las limitaciones de este tipo de cálculo es que no es muy adecuado para crear efectos de umbra y penumbra, lo cierto es que uno de los primeros métodos de computar la penumbra se llevó a cabo por medio de una extensión de este algoritmo, propuesta por Nishita y otros autores en

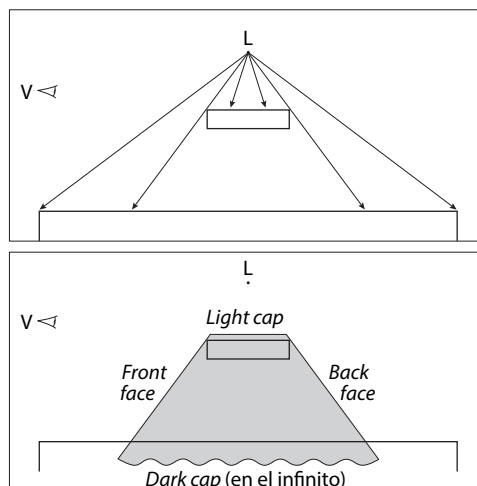


Figura 2.10 Cálculo de sombras por volumen de sombra (*shadow volume*).

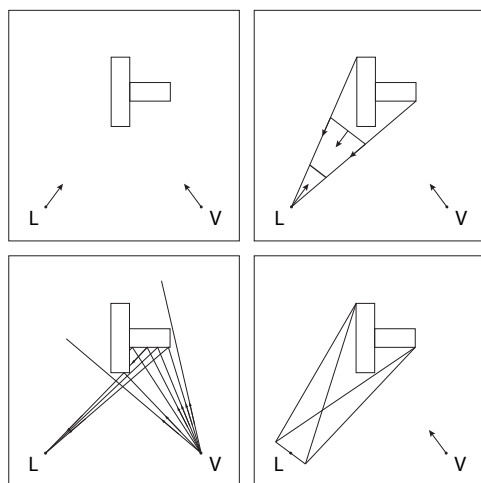


Figura 2.11 Cálculo de sombras por (a) mapa de sombras, (b) ray trace, (c) sombras de área.

1983, donde se generaban dos volúmenes de sombra a partir de los dos extremos de una luz lineal. Si un punto quedaba incluido en los dos volúmenes es que estaba en umbra, y si solo estaba en uno es que estaba en penumbra, y su intensidad se computaba a partir de una interpolación entre la región de umbra y la región iluminada. La definición del volumen requería un algoritmo de recorte poligonal complejo que, entre otras cosas, limitaba su aplicación a poliedros convexos.

Es un sistema poco utilizado en programas corrientes de simulación por lo que en los capítulos de técnicas y ejemplos nos centraremos en los otros dos, principalmente en el tercero.

### Sombras por mapa (*shadow map*)

*Mapa de sombras* (*shadow maps*, también denominados *depth map shadows*) es una técnica presentada por Lance Williams en 1978 y que se ha utilizado extensamente en todo tipo de aplicaciones de simulación desde entonces. Al igual que la anterior, pero en mayor medida, se ha beneficiado extensamente de la incorporación a las tarjetas gráficas de *buffers* especializados que permiten almacenar la in-

formación de profundidad y procesar el cálculo por hardware. Supuso una mejora importante con respecto al algoritmo de Crow pues permitía calcular con igual facilidad las sombras de objetos con superficies curvas, de ahí el título del artículo original (véanse las referencias).

La idea básica, que se ilustra en la figura 2.12, consiste en capturar la escena desde la posición del punto de luz, con lo que se obtienen los contornos que separan las zonas de luz de las de sombra. Para saber qué objetos están en sombra, se almacena la distancia desde la luz al objeto más cercano. Esta información se almacena en una zona de memoria especial denominada *z-buffer* porque corresponde a la coordenada Z de la escena desde el punto de vista de la luz. Al hacer un *render*, el cómputo de la iluminación se suprime más allá de estas distancias, lo que ahorra tiempo de cálculo. Por otro lado, esta información, que guarda los contornos de la sombra, es reproyectada sobre la escena desde el punto de vista del observador.

Cuanto mayor sea la resolución con que se ha captado la escena mayor será la precisión del resultado y viceversa. Por otro lado, cuanto mayor sea esta resolución, mayor será el coste de memoria y el tiempo de computación. Hay que ponderar lo que se desea y, en general, utilizar valores bajos en fase de pruebas y aumentarlos al final, si bien estas recomendaciones tienen mucha menos importancia en la actualidad que la que tenían hace años. Un valor, todavía dado por defecto en algunos programas, de 512, es bajo y habrá que aumentarlo probablemente hasta 1024 o 2048 para obtener resultados adecuados. Pero solo la práctica puede ayudar a decidir el mejor valor. Las ventajas principales de este tipo son que permite obtener sombras de bordes más o menos suaves, que solo se procesan una vez (excepto para objetos animados), por lo que es más rápido. Las desventajas son que utiliza más memoria ( $4 \times n \times n$  bytes siendo  $n$  el tamaño en píxeles: un tamaño de 4096 supondría  $4096 \times 4096 \times 4$  bytes o 64 Mb) y que no reconoce objetos con transparencia o mapas de opacidad. Al igual que ocurre con las texturas, es recomendable utilizar tamaños que





sean potencias de 2 (512, 1024, 2048...) pues las tarjetas gráficas están optimizadas para estos tamaños. Pero, como decía, los dos inconvenientes citados han dejado de serlo debido al avance en potencia de las tarjetas gráficas. Y si se utilizan ciertas variantes (como el *mental ray shadow map*) también se procesan transparencias.

El cálculo de sombras por mapa debe utilizarse en conjunción con luces focales que limitan el cómputo al área del foco (cónico o cilíndrico). Y cuanto más centrado sobre el objeto esté el foco, más preciso será el cálculo pues la resolución del mapa se aplicará a la zona que interese.

Excepto en casos en que realmente se necesite, no se deben utilizar sombras con luces omnidireccionales (puntuales) que calculan las sombras en todas direcciones pues puede suponer una carga importante e inútil para el motor de *render*.

El cálculo por mapa de sombras ha sido uno de los métodos de cálculo más utilizados. Pero a medida que ha aumentado la potencia de ordenadores y tarjetas gráficas está siendo substituido por el trazado de rayos con luces de área.

### Sombras por trazado de rayos (*ray traced shadows*)

Esta técnica fue introducida por Arthur Apple en 1968 y fue, de hecho, el primer paso para comenzar a desarrollar una de las primeras técnicas de iluminación global que se verán más adelante, *ray tracing*. Pero al igual que ocurrió con esta misma técnica, no pudo generalizarse hasta que la capacidad de los ordenadores pudo estar a la altura del gran número de cálculos requeridos.

El algoritmo se basa en enviar rayos trazadores desde la posición de la cámara, que rastrean la escena calculando las intersecciones con los objetos. Desde cada intersección se envían rayos hacia la luz. Si este nuevo rayo encuentra otra superficie en su camino quiere decir que el punto está en sombra. Y si no encuentra nada es que está iluminado.

Su resultado es más exacto y produce sombras de contornos nítidos, lo que puede ser deseable o no, según los casos. En general, se utiliza este método para sombras externas en condiciones de iluminación intensa. Las ventajas y desventajas son más o menos las inversas que el anterior. Utiliza menos memoria y solo se procesa una vez. También incluye menos parámetros de ajuste por lo que

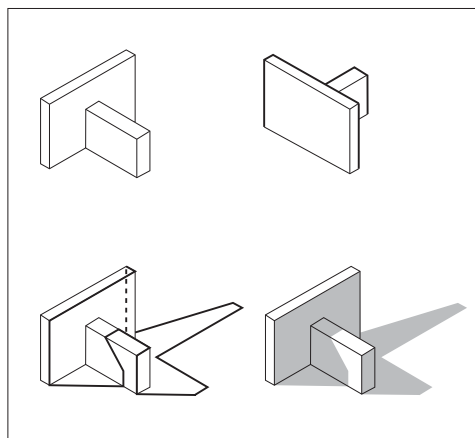


Figura 2.12 Cálculo de sombras por mapa de sombras. En (a) se muestran los objetos desde el punto de vista de la luz. En (b) se muestran los contornos de lo que ve la luz desde el punto de vista del observador. En (c) se muestran las sombras resultantes.

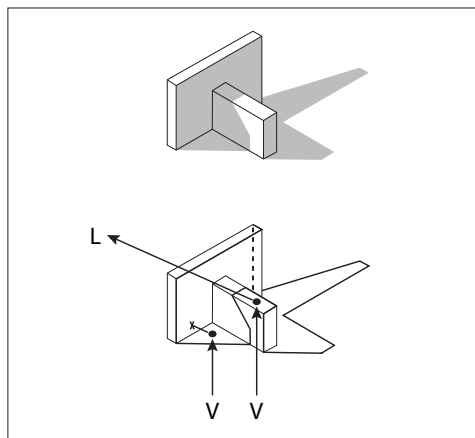


Figura 2.13 Cálculo de sombras por ray trace.





su puesta en acción es más sencilla. Pero el tiempo de cálculo es mayor y no permite obtener sombras de bordes suaves. Ha habido un gran número de mejoras técnicas dirigidas en todos los casos a aumentar la velocidad del proceso.

Otra ventaja importante a favor de las sombras por trazado de rayos es que pueden computar las sombras y el color arrojados por objetos semitransparentes o translúcidos algo que, en principio, no es posible con los métodos generales de sombras por mapa.

Su desventaja principal es que producen contornos muy duros que pueden ser adecuados para exteriores pero casi nunca lo serán para interiores. Las sombras por mapa

tienen más controles para suavizar los bordes.

Pero esta limitación principal deja de serlo si se utilizan sombras de área.

## Sombras de área

Las sombras de área se pueden obtener de dos modos: utilizando una luz puntual pero que cuente con un recurso específico denominado *area shadow* o, preferiblemente, utilizando una luz extensa, una luz de área que no necesite recurrir a este recurso. Y es preferible utilizar luces de área, entre otras cosas porque hay programas (como *mental ray*) que no incluyen dicho recurso.

En el primer caso, se utiliza un algoritmo que simula el área que tendría la luz si fuera extensa, aunque no lo sea.

En el segundo caso, se pueden utilizar tanto mapas de sombra como rayos trazadores. Pero los mejores resultados, aunque sean algo más lentos, se obtendrán con sombras por *ray trace* pues con esto se tendrá lo mejor de ambos mundos: la precisión de cálculo y la obtención de zonas de penumbra que se derivan naturalmente de la propia extensión de la luz.

En cualquier caso, deben utilizarse con alguna precaución pues aumenta considerablemente el tiempo de cálculo, aunque utilizan poca memoria.

## Sombras suavizadas por luces secundarias y por cálculos de iluminación global

Todos los métodos anteriores son métodos de cálculo que se basan en la iluminación directa. Recordemos que, en principio, con estos métodos se producirían sombras completamente negras. Y en ningún escenario real ocurre algo así: las sombras se aclaran al quedar iluminadas por objetos cercanos.

Este efecto puede conseguirse, utilizando luces estándar, por medio de luces secundarias que no arrojen sombras, dirigidas expresamente a las zonas de sombra, o bien por medio de sistemas de cálculo de iluminación



Figura 2.14 Ejemplos de cálculo de sombras por mapa de sombras, ray trace y sombras de área

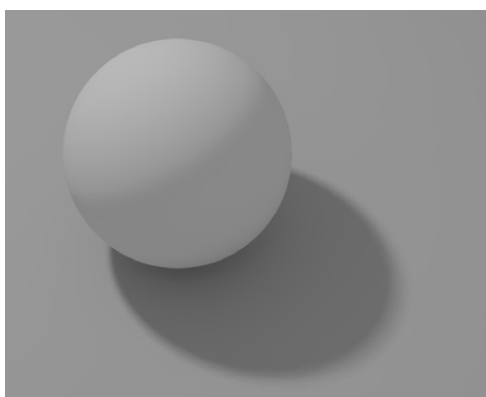
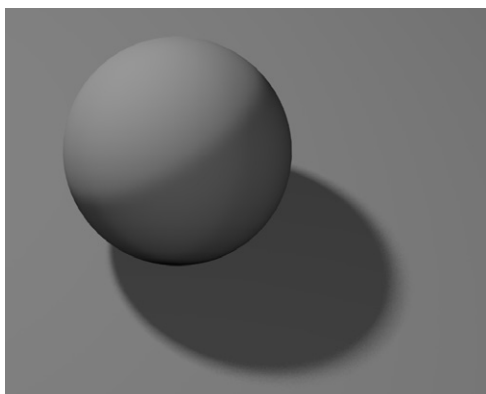
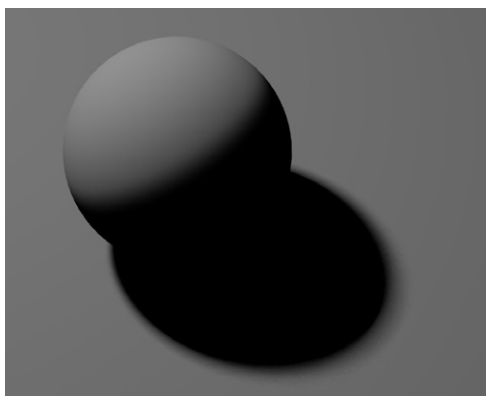


Figura 2.15 a) Sombras generadas por luces directas en sistemas locales; b) Las mismas sombras aclaradas por luces secundarias; c) Sombras generadas por sistemas de iluminación global.

global que computen este efecto. El primer caso es artificioso y difícil de controlar pues al iluminar las zonas en sombra también iluminamos las zonas en luz con lo que se altera la distribución original: su única (pero importante) ventaja es que el cálculo es mucho más rápido.

En el segundo caso obtendremos resultados realistas, con zonas de transición suave entre umbra y penumbra. Pero para ello se necesitan sistemas de iluminación avanzada.

### Sombras de objetos transparentes

Las sombras calculadas por mapa de sombra no pueden, en general (hay un tipo utilizado por *mental ray* que sí lo permite), simular las sombras arrojadas por objetos transparentes pues no tienen medios para saber cuáles son las propiedades del objeto ya que solo almacenan su posición geométrica.

En estos casos, es necesario utilizar sombras de *ray trace* que sí pueden computar las propiedades superficiales del objeto. Veremos ejemplos de estos casos en el capítulo 5.

### Otros tipos de sombras. Luces sin sombras. Luces negativas

Las luces virtuales hacen cosas que no es posible hacer con las luces reales. Esto abre múltiples posibilidades que pueden dar lugar a efectos seductores y originales, y también a aberraciones de todo tipo, por lo que deben utilizarse con prudencia. Los principales efectos son los siguientes.

Se puede hacer que una luz no arroje sombras desactivando esta opción. Este recurso es muy útil y se utiliza con frecuencia pues permite ajustar la intensidad y la distribución de la iluminación sin que aparezcan sombras indeseadas.

Se puede hacer que una luz no afecte a un determinado objeto que, en consecuencia, tampoco tendrá sombras. Esto puede hacerse mediante un parámetro especial que permite seleccionar objetos de la escena y excluirlos del cálculo.

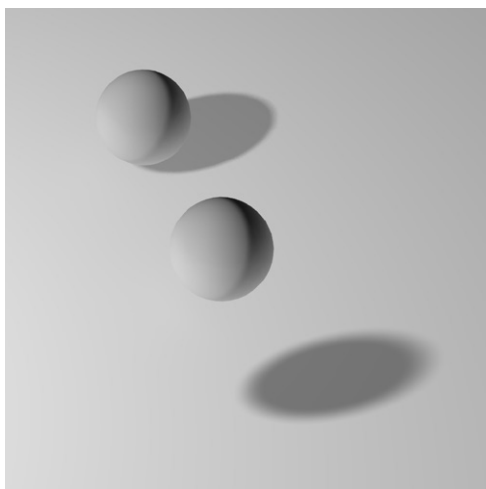


Figura 2.16 Exclusiones y luces negativas. La primera luz ilumina a la primera esfera y excluye a la segunda. La segunda luz ilumina a la segunda esfera, con las sombras desactivadas, y excluye a la primera. La tercera luz excluye a las dos esferas y arroja una luz de intensidad negativa.

Se puede hacer que una luz estándar arroje una luz oscura, lo que equivale a crear una sombra autónoma, dando al multiplicador un valor negativo. Este efecto, que ya se ha avanzado anteriormente, da lugar a efectos interesantes que pueden utilizarse para simular la atenuación por oclusión parcial con sistemas de iluminación local, para crear efectos similares a los de iluminación global. Un recurso bastante artificioso, pero muy eficaz para simular la leve zona oscura que un objeto deja a su alrededor al apoyarse en el suelo, es crear una luz en su interior, que excluya al propio objeto y que arroje una luz negativa muy suave, de bordes difusos sobre la base. Lo mismo puede hacerse para oscurecer una esquina sin tener que recurrir a iluminación global o algoritmos de oclusión parcial. Deben desactivarse las sombras y el componente especular de la luz.

También se puede *modificar la intensidad y el color de la sombra propia* mediante parámetros especiales que afecta a la densidad de la sombra, un recurso que se utilizaba cuando los sistemas de iluminación global eran prohibitivos.

Por último, también es posible crear *objetos que arrojen sombras pero sean invisibles*. Esto puede ser útil para controlar las sombras de figuras recortadas, más aún en el caso de figuras con planos cruzados para que sean visibles desde diferentes puntos de vista. En estos casos, puede ser útil crear dos objetos en el mismo sitio, uno visible, que no arroje sombras, y otro invisible, que arroje sombras. En 3ds Max, por ejemplo, todo lo que hay que hacer es editar las propiedades del objeto (*Botón Derecho Ratón/Object Properties*) y desactivar la opción *Visible to camera*. Al hacer un *render* la sombra será visible pero el objeto no. Véanse los ejemplos que se dan en el libro sobre simulación de materiales para simular figuras recortadas o árboles con sistemas de partículas.

## 2.5 Objetos virtuales. Los objetos virtuales como moduladores de luz

Al igual que en el capítulo anterior, este breve subcapítulo pretende recordar que sin objetos no hay escenarios, ni reales ni virtuales. Y que los objetos virtuales, como los reales, son también fuentes de luz, emisores secundarios.

Pero esto solo es así en los sistemas de iluminación avanzada o iluminación global. En los sistemas simples, de iluminación local, cada objeto *recibe* un color y una intensidad de iluminación pero no la reenvía a su alrededor. El aspecto artificioso de las escenas de este tipo se debe, precisamente, a que falta la unidad característica de una escena real en donde la energía lumínica se modula armoniosamente por la contribución ponderada de las reflectancias de todos los objetos, pequeños o grandes, oscuros o claros.

Hay que insistir también, una vez más, en que “artificioso” es un término pariente de “arte”, y una escena con un carácter más abstracto, menos real, puede ser un modo más eficaz de representar un espacio. La si-

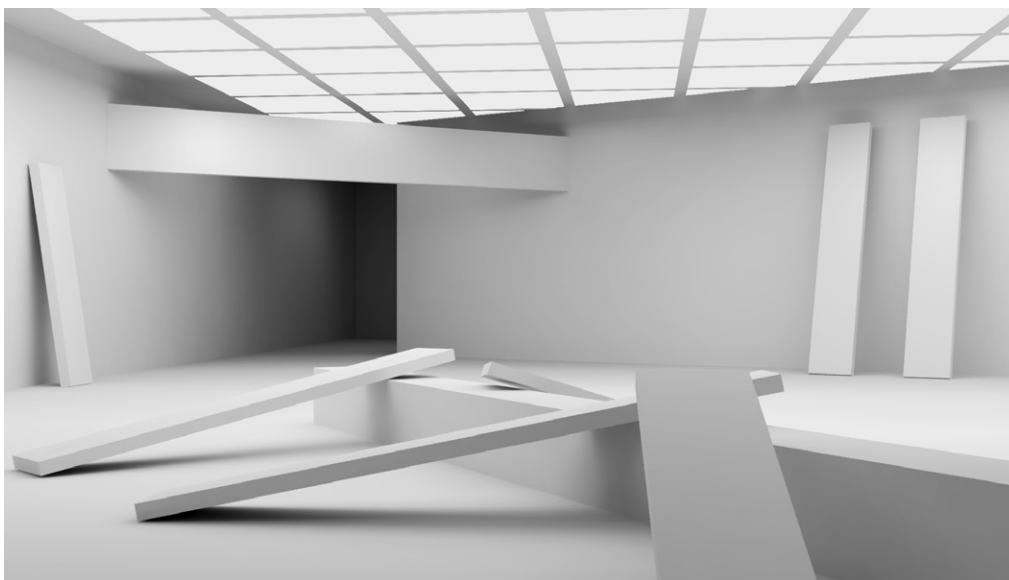


Figura 2.17 Escena con varios objetos que modulan la luz de diferentes modos.

mulación local tiene importantes ventajas, si se sabe utilizar con habilidad, algo que no hay que perder de vista.

En cualquier caso, como veremos en el capítulo correspondiente a los sistemas de iluminación avanzada, los métodos de cálculo, en estos casos, dependen de los valores de reflectancia de los objetos. Estos valores de reflectancia se pueden simular de diversos modos pero, en general, dependen de una función, la “función de distribución bidireccional de las reflectancias”, BRDF (*bidirectional reflectance distribution function*), que está incorporada en versiones simplificadas a todo tipo de materiales y en versiones más complejas a materiales avanzados. En el libro sobre simulación de materiales se proporciona amplia información sobre esta función y su uso en la práctica.

En el contexto de este libro puede bastar con recordar que los sistemas de iluminación avanzada dependen de los rebotes de la luz en los objetos, y que estos rebotes dependen a su vez de que se hayan especificado propiedades de reflectancia a los materiales de la escena.

Dado que aquí se prescinde del color y otras propiedades de materiales, podemos quedarnos con la idea de que esta redistribución de la luz, que corre a cargo de los materiales, depende del nivel de gris correspondiente al color local del material y de la mayor o menor reflectancia asignada a los materiales. También en este caso ampliaré la información con ejemplos concretos.

## 2.6 Ojos/cámaras virtuales

Como ya he dicho al comienzo de este capítulo, la analogía entre escenarios reales y virtuales se quiebra, en cierto modo, cuando hablamos de ojos o cámaras virtuales. Para ver a través de un ojo virtual necesitamos un ojo real. Y el ojo virtual, pasivo, que no es sino una proyección geométrica, tiene poco que ver con el ojo real, activo, con su extraordinaria capacidad de adaptación, anticipación e interpretación.

En cualquier caso, es necesario entender bien los pasos involucrados en este proceso. Lo que sigue no es sino una introducción a los

pasos de un recorrido complejo al que habrá que volver con más detalle en el capítulo 4.

Una escena virtual se visualiza a partir de una cámara virtual. Y una cámara virtual no es sino la especificación de una proyección que implica un cambio de coordenadas junto con otras propiedades.

Las propiedades geométricas principales de una cámara virtual son: a) su posición en la escena, b) su orientación, la dirección a que apunta, c) su relación de aspecto, d) su ángulo de visión. La figura 2.18 condensa estas cuatro propiedades.

A partir de estos datos, podemos obtener mecánicamente una proyección 2D de la escena 3D. Pero en esta proyección ocurren más cosas.

En primer lugar, habremos pasado de un escenario 3D, en donde cualquier punto de un objeto puede ser definido con una precisión muy alta, a un escenario 2D, en donde solo hay un número limitado de puntos, cada uno de los cuales corresponde a una zona o a varias zonas superpuestas que se proyectan y confunden sobre la zona proyectada.

En segundo lugar, el color final de cada uno de esos puntos 2D, de cada píxel resultante, no corresponde a nada que “esté” en la escena sino a un cálculo complejo en el cual,

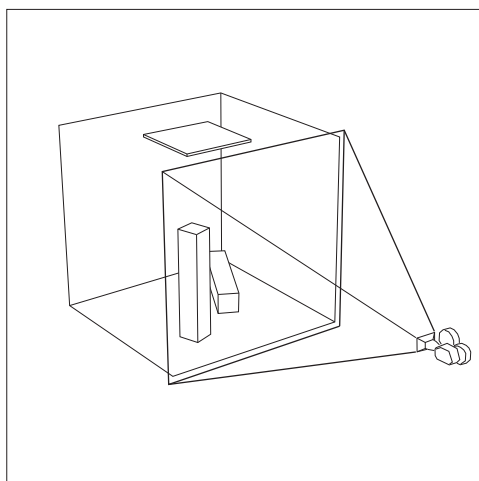


Figura 2.18 Cámaras virtuales.

a partir de la información geométrica, más las propiedades definidas para cada material, más las modificaciones de esas propiedades dadas por el sistema de iluminación, más los ajustes introducidos por la configuración de cálculo escogida, más los ajustes añadidos por los filtros y otros mecanismos de corrección de defectos, más otros ajustes que iremos viendo más adelante, a partir de todo esto, se genera el color particular del píxel particular.

En tercer lugar, el color así obtenido no es aún un color visible sino un color que se envía a un “marco de memoria de visualización”, al *frame buffer* del sistema, preparado para ser volcado sobre el dispositivo de salida. Es decir, a una “imagen” invisible preparada para ser visible. Y en este vuelco tiene lugar una última transformación por la cual los colores que hemos obtenido se adaptan a las características concretas del dispositivo de salida que, para no complicar más esta exposición, vamos a suponer que es un monitor corriente.

Y en ese momento, lo virtual se transforma en real para ser contemplado, a su vez, por un ojo humano (real).

En los programas de simulación se utilizan dos tipos de cámaras: cámaras básicas y cámaras físicas.

Una *cámara básica* es un objeto auxiliar que simula la posición de una cámara en la escena, pero cuya función principal se reduce a especificar el punto de vista desde el que se llevará a cabo una proyección en perspectiva. Estas cámaras básicas incluyen muy pocas propiedades. La principal es, a) la especificación del ángulo de la pirámide de visión o campo de visión, *FOV* por sus siglas en inglés (*field of view*).

Y según los diferentes programas, pueden incluir también las siguientes: b) la especificación de planos de corte, *clipping planes*, que permiten limitar la proyección entre dos planos, anterior y posterior y c) la especificación de rangos de entorno, distancias a partir de la cámara que pueden asociarse a efectos atmosféricos.



Una *cámara física* es un objeto más complejo que, además de los controles propios de las cámaras básicas, incluye propiedades características de las cámaras reales. La principal es, a) el control de exposición, sea mediante un valor global, EV, sea mediante parámetros separados para la apertura del diafragma o número  $f$  (*f-stop*), la velocidad de obturación (*shutter speed*) y la velocidad de la película (*film speed*) medida en general en valores ISO.

Y según los diferentes programas, pueden incluir también las siguientes: b) balance de blancos, c) control del viñeteado, d) control de la profundidad de campo, DOF por sus siglas en inglés (*depth of field*), e) corrección de las fugas verticales y f) desenfoque de movimiento (*motion blur*).

Algunos programas mantienen alguno de los controles anteriores de modo separado. Por ejemplo, 3ds Max tiene un único tipo de cámara (que incluye también controles de profundidad de campo) y el control de exposición se activa desde un módulo independiente. También incluye un operador especial que puede añadirse a la cámara para la corrección de verticales, y otros controles adicionales independientes para el desenfoque de movimiento.

## 2.7 La formación de imágenes virtuales. Límites geométricos

### Límites geométricos. Capacidad de resolución de dispositivos virtuales

Una de las primeras cosas que necesitamos saber antes de iniciar un proyecto de simulación es cómo va a terminar. Es decir, si el destino de la imagen final es ser impresa en pequeño o gran tamaño, ser proyectada en un monitor de pequeño o gran tamaño o ser enviada por red. Necesitamos, por tanto, saber qué *resolución* final deberá tener la imagen.

El término *resolución* es muy ambiguo y se presta a numerosas confusiones. Así que co-

menzaré por precisar que cuando hablamos de *resolución* podemos estar hablando de varias cosas diferentes.

El origen del término se basaba en la capacidad de un dispositivo para discriminar detalles o, más exactamente, la capacidad para distinguir (“resolver”) dos puntos muy próximos que pueden aparecer como un punto único. Concretamente, se utilizaba y se utiliza en astronomía para referirse a la capacidad de un telescopio para distinguir dos estrellas que, para un telescopio “de menor resolución” eran una sola estrella, tal como se muestra en la figura 2.19.

En el caso de las imágenes digitales, que están compuestas por puntos discontinuos pero que se presentan como líneas o superficies continuas, se utiliza de dos modos muy diferentes de los que derivan la mayor parte de los malentendidos, propiciados también por el hecho de que muchos programas utilizan indistintamente ambos términos. Para evitar esta confusión distinguiré, cuando sea necesario, entre *resolución propia* y *resolución gráfica*.

Por *resolución propia* de una imagen, entendemos la cantidad de detalle que incorpora. Y esto depende de las dimensiones de la imagen, medidas en píxeles. Para simplificar se expresa corrientemente por su producto en megapíxeles (MP). Así, una imagen de 1.600 x 1.200 píxeles tendría una resolución de 1.920.000 píxeles o 1,92 MP. Y, a la inversa, para un formato de imagen de 3/4, una resolución,  $R$ , de, por ejemplo, 12 megapíxeles, equivaldría a una imagen con una anchura de 4.000 píxeles (anchura = raíz cuadrada de  $4/3R$ ).

También debe tenerse en cuenta que la resolución propia solo tiene sentido si conocemos el contexto en que se va a utilizar la imagen y, entonces, se puede expresar en  $u/px$ , donde  $u$  son unidades que dependen de este contexto (milímetros, centímetros, metros, etc.) y  $px$  son píxeles. Si, por ejemplo, tenemos una imagen del rostro de una persona y hacemos una ampliación acercándonos a sus ojos podríamos, o no, distinguir su iris (que mide, aproximadamente 1 cm de diáme-



tro) según la resolución de la imagen. Naturalmente el que la resolución sea “suficiente” dependerá también en este caso de para qué queremos utilizar la imagen. Si solo queremos distinguir los rasgos básicos nos bastará, para este ejemplo, con unos 0,25 cm/px o menos. Pero si queremos hacer un diagnóstico de posibles enfermedades por medio del iris necesitaremos, seguramente, 0,0025 cm/px o más.

La resolución tiene, por tanto, una importancia considerable en aplicaciones prácticas: una fotografía aérea será útil, en aplicaciones orientadas a la arquitectura y urbanismo, según su resolución. Si la imagen tiene 1.000 píxeles de anchura y representa un área de 100 metros de ancho, su resolución será de 0,1 m/px (100/1.000) o de 10 cm/px. Que esto sea “suficiente”, o no, dependerá de para qué la queremos. Servirá si solo nos interesa el volumen global de un edificio. Pero será insuficiente si queremos saber de qué tipo son sus ventanas.

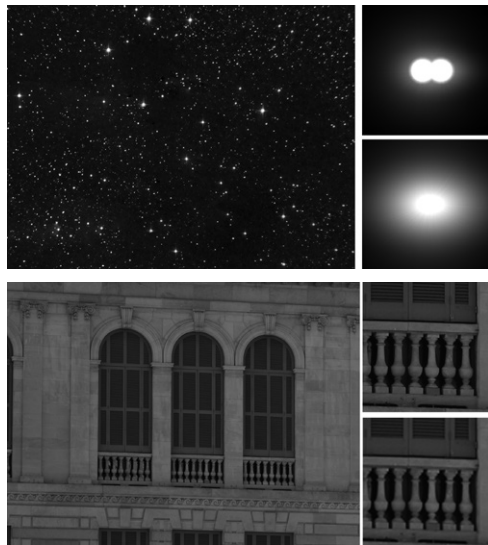


Figura 2.19 Resolución propia: a) La resolución de un telescopio permite distinguir dos estrellas donde aparentemente solo había una, b) La mayor o menor resolución de una imagen, indicada en píxeles por unidades reales, permite discriminar detalles tales como las características de la carpintería de una ventana.

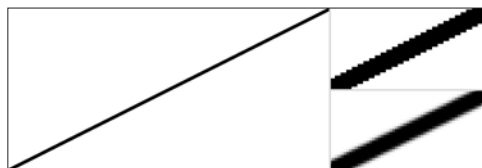


Figura 2.20 Resolución gráfica. La mayor o menor resolución de una imagen, expresada en píxeles por unidades de tamaño de salida de la impresión gráfica, permite contar con la certeza de que no se van a apreciar las discontinuidades características de las imágenes digitales.

Por resolución gráfica de una imagen entendemos la misma relación anterior, es decir, la relación entre los píxeles de una imagen y las unidades de medida corrientes pero, en este caso, referidas no a “la realidad” sino al tamaño del papel o del dispositivo sobre el cual se va a imprimir o proyectar. En este caso, el tamaño deberá ser tal que las líneas inclinadas aparezcan como líneas continuas (en lugar de lo que realmente son, líneas escalonadas discontinuas, píxeles yuxtapuestos de igual tono), y las superficies con degradados suaves aparezcan igualmente como superficies continuas (en lugar de lo que realmente son, píxeles yuxtapuestos de tonos diferentes).

La resolución “adecuada” para este caso no depende solo del número de píxeles de la imagen sino también de las características del medio de salida. Si la imagen está destinada a ser proyectada sobre una pantalla de televisión o un monitor, la irradiación de los puntos hace que los defectos queden disimulados y que se necesiten muchos menos puntos para obtener un resultado satisfactorio. Como hemos visto en el capítulo anterior, la imagen derivada de un punto tiene un tamaño que depende no solamente de la capacidad puramente geométrica de nuestros receptores retinianos sino de otros factores, entre los cuales hay que incluir la difracción y la función de dispersión puntual o lineal.

Una trama de puntos luminosos, de unos 0,25 mm de diámetro, equivalentes a una resolución de unos 28 ppc (puntos por centímetro) o 72 dpi (dots per inch), que son precisamente las características de un monitor más





o menos corriente, será indistinguible a partir de unos 50 cm de distancia pues se fundirá en un continuo. Pero si la imagen se imprime como puntos negros sobre un fondo blanco, ocurre lo contrario: la luminosidad del fondo hace más visibles los defectos de las líneas impresas. Los datos experimentales que hemos visto en el capítulo anterior, aplicados a estos casos, nos dicen que se requieren al menos 120 ppc (300 dpi) para que las líneas inclinadas y las superficies se vean continuas si se imprimen en negro sobre blanco.

La ambigüedad del término se complica todavía algo más por las diversas especificaciones que se encuentran en diferentes contextos. Solo en la literatura anglosajona (para no entrar en otra complicación derivada de las diferentes traducciones de estos términos) se encuentran tres denominaciones para indicar casi lo mismo: dpi, ppi y lpi.

Se denomina *dpi* (*dots per inch*) o “densidad de puntos”, a la resolución de una imagen digital con respecto al número de puntos que pueden colocarse en línea al enviarlos a su salida prevista, generalmente por impresora. Este es el término más corriente y el que utilizaremos a lo largo de este libro.

Se denomina *ppi* (*points per inch*) o “densidad de píxel” (*pixel density*) a la resolución nominal de una imagen digital con respecto a su salida prevista o, dicho de otro modo, a la resolución mostrada o *display resolution*. El término es prácticamente sinónimo de dpi, hasta el punto de que se usa uno y otro en diferentes contextos o por diferentes programas, pero enfatiza el hecho de que un píxel puede no corresponder a un punto. Esto es notorio en el caso de un monitor en el cual podemos cambiar la resolución modificando el número de píxeles que se procesan para un mismo tamaño físico.

Se denomina *lpi* (*lines per inch*) o *lpm* (*lines per mm*) o “líneas por pulgada” o “líneas por milímetro” o “frecuencia lineal”, a la resolución de una imagen especificada en términos del número de líneas que pueden imprimirse por unidad (pulgadas o milímetros) cuando se usa impresión por semitonos. El término es,

de nuevo, equivalente a los anteriores, pero por razones de continuidad con la tradición (la calidad se especifica en relación al número de semitonos que podían incluirse por línea) se utiliza más en el mundo de la fotografía y de la impresión artística o comercial. En fotografía se utilizan los valores de referencia siguientes (datos tomados de un manual reciente de Kodak: el primer valor indica las líneas por mm y el segundo, la clasificación ISO-RP): 50 lpi o menos: “bajo”, 63 a 80: “medio”; 100 a 125: “alto”; 160 a 200: “muy alto”; 250 a 500 “extraordinariamente alto”. También se usa, en este mismo contexto, la denominación *spi* (*samples per inch*) para referirse a la resolución de una imagen escaneada, para imprimir, que se considera que debe tener una resolución del orden de 1.5 a 2 veces la de impresión es decir que si el valor de salida fuese 100 lpi el de entrada debería ser entre 150 y 200 spi.

Nos basta con recordar los primeros valores dados al comienzo: *dpi* o *ppc*. Pero he citado los otros para recordar que no son los únicos que se utilizan en la práctica y que pueden aparecer en diferentes contextos.

También hay que recordar que la resolución por sí sola no es suficiente para determinar la calidad potencial de una imagen. En el caso de dispositivos de captación, como una cámara digital, a partir de un punto determinado, el punto límite dado por la difracción, un aumento en la resolución no mejorará la calidad de la imagen. Esto es debido a que el tamaño de la apertura de una cámara digital está limitado y este límite afecta a la resolución de la imagen. De hecho, cuanto más pequeño es el tamaño de la apertura mayor es el efecto de este límite. Me remito de nuevo al capítulo anterior, donde ya se ha analizado este fenómeno al que los fotógrafos se refieren como “círculos de confusión”, y que está relacionado con los discos de Airy, y que designa los valores límite por debajo de los cuales es innecesario aumentar el enfoque pues no sería posible percibir detalles.

La conclusión de todo esto es que tan solo los valores experimentales, dirigidos por análisis teóricos, pueden orientarnos adecuadamente.



te sobre las resoluciones adecuadas y que, por añadidura, estos valores experimentales deben acabar de afinarse en relación con los dispositivos concretos que vayamos a utilizar.

Las impresoras modernas se anuncian con resoluciones del orden de los 1200 o 1440 dpi o más. Sin embargo, cualquier profesional de artes gráficas nos dirá que no merece la pena imprimir una imagen a una resolución mayor de 400 dpi, y a menudo es suficiente con 300 o menos. ¿Por qué se utilizan entonces resoluciones mucho más altas? La razón es que los tonos se generan mediante agrupaciones de celdas

El problema con que nos encontramos habitualmente, si queremos ajustar lo mejor posible la resolución de nuestras imágenes a las características de la impresora, es que no sabemos cómo se agrupan estas celdas porque los fabricantes no lo dicen.

Sería interesante saberlo para adecuar la resolución a esta agrupación. Dado que esta agrupación debe ser alguna de las que se dan en la tabla siguiente, parece claro que los valores están fijados y cualquier valor intermedio se modificará para ajustarlo a estos valores.

Celda	1.200 dpi	1.440 dpi
4 x 4	300	360
5 x 5	240	288
6 x 6	200	240
8 x 8	150	180

Sin embargo, esto no son sino indicaciones que apuntan al marco general en que se dan determinadas especificaciones y que no está de más conocer. En la práctica, teniendo presente este marco general, lo más recomendable será hacer pruebas con dispositivos y modelos concretos.

## 2.8 La formación de imágenes. Rangos dinámicos y escalas de valores

La segunda cosa que necesitamos saber antes de iniciar un proyecto de simulación es con qué calidad podemos contar. Entendien-

do en este caso por “calidad” algo muy concreto: la capacidad del dispositivo final de salida para reproducir el rango de intensidades que hayamos especificado para nuestra imagen final, así como el modo en que los diferentes controles dados por las configuraciones de cálculo de nuestro programa afectan a estos rangos de intensidades.

En el capítulo anterior también hemos visto cómo el sistema visual humano es capaz de captar diferencias de intensidad dentro de determinados rangos dinámicos que corresponden a diferentes situaciones habituales. También he introducido el tema de la construcción de escalas de intensidades.

Los dispositivos virtuales tienen diferentes rangos dinámicos y diferentes modos de distribuir los valores de luminancia en estos rangos. Esto plantea problemas importantes que no están resueltos. Debe recordarse otra vez, para entender la importancia de esta sección, que los programas de simulación de iluminación hacen fundamentalmente dos cosas. Primero, calculan la distribución de valores de radiancia en la escena que se está simulando, a partir de las fuentes de luces especificadas y los valores de reflectancia y otras características de los objetos de la escena. Este cálculo es un cálculo físico, un cálculo de valores precisos que se registran de modo lineal. Segundo, redistribuyen estos valores para que aparezcan de un modo adecuado en un dispositivo de salida que va a ser contemplado por un observador humano. Aunque, como veremos más adelante, hay un gran número de artículos, libros y estudios sobre lo primero, sobre el cálculo físico, hay menos estudios sobre lo segundo. Y resulta que, como sabe cualquier usuario experimentado en programas de simulación, después de cálculos extraordinariamente sofisticados para obtener los valores físicos, al final nos encontramos ajustando estos valores de salida (jugando con el control de exposición de 3ds Max, por ejemplo), a ojo. De ahí la importancia de entender con claridad los problemas involucrados.



## Rangos dinámicos de dispositivos artificiales

Como hemos visto en el capítulo anterior, el rango de luminancias que el sistema visual humano es capaz de percibir es muy amplio y varía según las circunstancias. Otro tanto ocurre con los dispositivos capaces de generar imágenes virtuales, dando al término “dispositivo” un sentido muy amplio que abarca tanto sistemas capaces de captar imágenes, como cámaras o escáneres y sistemas capaces de reproducirlas, como monitores, impresoras, hojas de papel o proyectores.

Cada uno de estos sistemas tiene su propio rango dinámico. Es muy importante entender cuál es este rango dinámico para saber qué podemos esperar de cada uno de los sistemas, y este apartado está dedicado a describir los principales.

Tal como se comentará más adelante, no hay que perder de vista que una cosa son los rangos dinámicos nominales y otra distinta los reales. Como hemos visto en el capítulo anterior, podemos asumir 100:1 como un factor de contraste de referencia suficiente para la gran mayoría de las aplicaciones prácticas, y un umbral de discriminación de un 1 %. Y debido a fenómenos como la dispersión lumínica en el ojo, que también se ha mencionado en ese capítulo, los rangos máximos que podemos discriminar no sobrepasarán en ningún caso los 150:1. Pero hay que tener igualmente presente que la calidad de una imagen depende de su capacidad de registrar rangos que no podemos percibir directamente pero sí indirectamente.

Los valores de luminancia en interiores se sitúan corrientemente en un rango de 100 a 1.000 cd/m<sup>2</sup>. Y los monitores están limitados a un rango real que estaría situado en la práctica corriente en valores que irían de 1 a 30 cd/m<sup>2</sup>, aunque puedan alcanzar luminancias bastante mayores y sus rangos nominales sean de 100:1 o de 300:1 (y los valores nominales dados por los fabricantes mucho mayores aún). Los algoritmos de redistribución de tonos que se utilizan con formatos HDR, que

se verán más adelante, consideran que las luminancias máximas reales que se alcanzarán en un monitor no serán superiores a los 100 cd/m<sup>2</sup>, y pueden ser de la mitad en la mayoría de los casos.

## Rangos de dispositivos de entrada. Cámaras digitales. Escáneres

Los principales dispositivos de entrada son los escáneres y las cámaras digitales.

**Cámaras.** Las cámaras digitales, al igual que los escáneres, se basan en sensores. Los sensores de las cámaras digitales están formados por detectores individuales, fotodiodos, que acumulan fotones hasta alcanzar un punto de saturación determinado, el correspondiente a la máxima intensidad de luz que pueden recibir. Un sensor es un tipo especial de transductor (*transducer*), un dispositivo que transforma un tipo de energía en otro. Los sensores se pueden clasificar por el tipo de energía que detectan. Los sensores utilizados en escáneres y cámaras pertenecen a un amplio grupo de sensores lumínicos (*light sensors* o *photodetectors*), que incluye detectores ópticos de diversos tipos, células fotovoltaicas, fotodiodos y, entre otros, CCD y APS/CMOS.

Un sensor CCD (*charge-coupled device*, dispositivo de carga acoplada) es un dispositivo que incluye básicamente una capa que reacciona a la luz y otra capa que transmite esta reacción. La imagen se proyecta por medio de una lente sobre la capa sensible a la luz. Esta capa incorpora una matriz de detectores que traducen la energía lumínica a una determinada carga eléctrica, proporcional a la energía recibida. Un circuito de control hace que la carga de cada condensador de capacidad (*capacitor*) transfiera su energía a su vecino. El último elemento de esta matriz envía su carga a un amplificador que convierte esta energía en un voltaje determinado. Repitiendo este proceso, la secuencia de intensidades se convierte en una secuencia de voltajes que se almacenan en la memoria del dispositivo. Los primeros CCD se produjeron

en 1969 en los laboratorios Bell, y puede decirse que con ellos nació la fotografía digital. Las primeras aplicaciones comerciales se pusieron en circulación en 1974, y a principios de la década de 1980, Sony lanzó las primeras producciones masivas. Un CCD corriente puede captar un rango dinámico de unos 60 a 70 dB, lo que equivale a unos 10 a 11,5 EV (ver los comentarios que se incluyen más abajo sobre el valor de exposición, EV, *exposure value*). Los mejores CCD pueden llegar a un rango de 78 dB, equivalentes a 13 EV. Pero esta tecnología no dará más de sí por lo que las cámaras de alto nivel utilizan otros tipos de sensores.

Las cámaras digitales avanzadas utilizan sensores denominados APS. Un sensor APS (*active pixel sensor*) es un sensor basado en tecnología CMOS (*complementary metal-oxide-semiconductor*) por lo que es también conocido como *sensor CMOS*. La tecnología CMOS utiliza un tipo especial de circuito integrado que se usa en muchos tipos de aplicaciones (microprocesadores, RAM estática...). Utiliza pares complementarios (de ahí el nombre) de semiconductores *metal-oxide* positivos y negativos. Los sensores de tipo APS incluyen un fotodetector en cada píxel junto con un amplificador. Los píxeles "activos" de este tipo de tecnología, a diferencia de los "pasivos" de los CCD, favorecen una mayor velocidad y mayor facilidad de producción. Se utilizan sobre todo en las cámaras incorporadas a los teléfonos móviles, en cámaras web y en cámaras digitales reflex.

La capacidad de captación de los sensores está relacionada con las características citadas y con su tamaño. Como puede apreciarse en la figura 2.21, estas diferencias de tamaño son muy importantes y están relacionadas con el rango dinámico de las cámaras digitales. Los sensores de las cámaras digitales SLR son de mayor tamaño y tienen también mayor capacidad de acumulación. En el otro extremo, el extremo inferior correspondiente a valores oscuros, los límites vienen dados por la capacidad de medición que, a su vez, de modo similar a lo que ocurre con los escáneres, queda limitada por el ruido, por la indeterminación de la señal.

Este *ruido* de fondo es debido a varios factores. Los principales son debidos, por un lado, a imprecisiones de lectura y, por otro lado, a la propia variación de impacto de los fotones. Ambos factores son imposibles de evitar. El número de fotones que llegan a un sensor es fluctuante y esto introduce una indeterminación que se traduce en ruido. La relación entre señal y ruido (*signal to noise ratio*) se define, en procesamiento de señales, como una desviación de la media estadística que es necesario corregir en cualquier caso por diversos medios, principalmente mediante el uso de

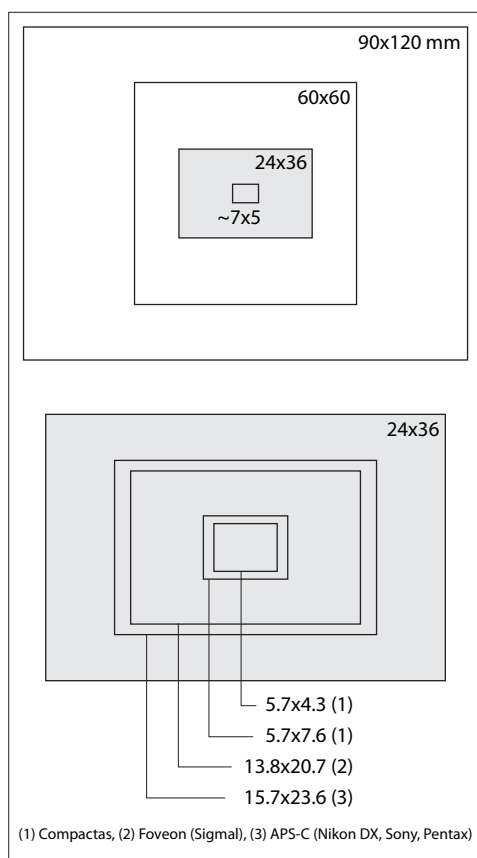


Figura 2.21 Comparación de tamaños de sensores tradicionales y sensores digitales.



filtros. La relación entre el rango dinámico y el ruido se expresa por la fórmula  $RD = 20 \times \log_{10}(\text{MaxS}/\text{MinS})$  donde RD es el rango dinámico en dB, MaxS, la señal máxima que se puede introducir en el sistema antes de que se sature, es decir, antes de que se recorte para igualarse al valor anterior, y MinS es la última señal que se puede introducir en el sistema antes de que pueda distinguirse de puro ruido. Por esta razón, el tamaño de los sensores es un factor de considerable importancia; aunque el ruido de fondo sea el mismo, un sensor de mayor tamaño recibe un mayor flujo de fotones durante un mismo tiempo de exposición. Y así, la relación entre los valores mayores y menores (la relación señal/ruido) aumenta.

Para reducir este ruido de fondo se han ensayado diversos métodos. El más obvio sería enfriar el sensor, pues como es bien sabido la agitación electrónica aumenta con la temperatura y disminuye al disminuir esta. Pero es igualmente obvio que no podemos pasear nuestras cámaras en un congelador y los sistemas de enfriamiento microscópico no parece que puedan aislarse con facilidad. Otros métodos más sofisticados permiten mejorar la capacidad de captación de los sensores. Los sensores CMOS, de hecho, no funcionan como los sensores CCD, que acumulan cantidades de luz en forma de cargas ligadas linealmente a la intensidad. Los sensores CMOS tienen una respuesta logarítmica, basada en transistores adicionales, que les permiten amplificar la señal de un modo similar a como funcionaría una película analógica tradicional, lo que supone mayor capacidad para almacenar información y, en consecuencia, su rango dinámico intrínseco también es mayor. Pero las variantes y la complicación de la descripción de estos y otros tipos experimentales rebasan el campo de este apartado. En cualquier caso, esto no es sino una indicación de que la evolución industrial apunta en esta dirección, lo que confirma que estamos en una segunda fase: las cámaras ya tienen suficientes megapíxeles como para producir resultados de resolución adecuada para un

gran número de aplicaciones. El siguiente reto es aumentar su rango dinámico.

El rango dinámico de las cámaras digitales se obtiene, como en todos los casos, por la relación entre la intensidad máxima y mínima que es capaz de captar. Este rango dinámico, por las razones citadas, es bastante inferior al de las cámaras tradicionales. Una película pancromática de grano fino puede captar hasta 4 órdenes de magnitud, un valor cercano al del ojo humano adaptado, en torno a los 10.000:1. Este es un valor potencial, correspondiente a una densidad de 4,0 y que permite a los fotógrafos manipular el negativo entre unos límites que no superarán una densidad de 3,0 (1000:1) y un negativo cuyo rango dinámico será aún inferior. Pero los sensores de las cámaras digitales de gama alta tienen valores inferiores de entrada. Aunque nominalmente un sensor típico pueda captar 12 bits, lo que supondría una gama dinámica de 4.096:1, en realidad las cámaras digitales no superan rangos de 1.000:1 y lo más corriente es que este valor esté en torno a 500:1.

En las cámaras, el rango se mide corrientemente en pasos de exposición. Como este es un parámetro muy importante que también se utiliza en los procedimientos de simulación visual que se verán más adelante, conviene añadir algo más sobre el valor de exposición.

El valor EV (*exposure value*), desarrollado hacia 1950 por grupos de trabajo alemanes y norteamericanos, equivale a una combinación de aperturas y velocidades que lleva a una progresión geométrica  $2^{EV}$ . Un valor de EV 0 corresponde a una apertura  $f/1,0$  y 1 segundo. La progresión se relaciona, en general, con la apertura y la velocidad mediante la fórmula siguiente (para una sensibilidad de negativo ISO 100):

$$2^{EV} = N^2/t$$

donde:

$$EV = \log_2 N^2/t$$

y donde  $N$  es la apertura del diafragma (el valor de  $f$ ) y  $t$ , el tiempo.

Si queremos evitar los logaritmos en base 2, dado que



$\log_2(n) = \log_{10}(n)/\log_{10}(2) = \log_{10}(n) \times 3.321$ , la fórmula anterior será equivalente a:

$$EV = 3,3219 \times \log_{10} N^2/t$$

Por ejemplo, para  $f = 16$  y  $t = 1/30$ , tendríamos que  $EV = 3,3219 \times \log(16^2 \times 30) = 3,3219 \times \log(256 \times 30) = 3,3219 \times 3,8853 = 12,9$ .

Así, pueden calcularse valores que equivalen a todas las combinaciones posibles entre aperturas y velocidades. Pueden encontrarse tablas que relacionan los valores de EV corrientes, entre 1 y 16, en cualquier manual de fotografía o haciendo una búsqueda por Internet. La tabla de la figura 2.22 da referencias genéricas para estos valores.

Si se utilizan diferentes valores de ISO hay que corregir la fórmula anterior. Pero como lo que nos importa es ver el sentido básico de este parámetro de referencia, me remito a un buen manual de fotografía.

Es posible, en principio, calcular la luminancia y la iluminación a partir de valores EV

si conocemos las constantes de calibración de una cámara determinada.

Para *fotómetros que miden la luz reflejada*, los valores de exposición se calculan según la siguiente fórmula (dada por la ISO 2720:1974):

$$2^{EV} = N^2 / t = L_s S / K$$

en donde  $N$  y  $t$  significan lo mismo que arriba,  $L_s$  es la luminancia de la escena ( $\text{cd/m}^2$ ),  $S$  la velocidad ISO y  $K$  la constante de calibración del fotómetro de reflexión (véase más abajo).

Para *fotómetros que miden la luz incidente*, los valores de exposición se calculan según la fórmula siguiente (dada por la ISO 2720:1974):

°en donde  $N$ ,  $t$  y  $S$  tienen el mismo significado que antes,  $E$  es la iluminancia de la escena (luxes) y  $C$ , la constante de calibración del fotómetro de incidencia. Es importante tener en cuenta que las situaciones pueden ser muy variadas pues se puede medir la intensidad “media” de la escena, la intensidad de una

EV	Escena característica
-4	Luz nocturna, media Luna
-3	Luz nocturna
-2	Luz nocturna, Luna llena
-1	Iluminación por luces artificiales poco intensas
0	Iluminación por luces artificiales poco intensas
1	Vela a 1 m de distancia
2	Edificios iluminados a distancia
3	Iluminación de monumentos o fuentes
4	Iluminación urbana nocturna suave
5	Iluminación urbana nocturna intensa
6	Interiores domésticos
7	Áreas de trabajo en oficinas
8	Interiores muy iluminados
9	Luces artificiales en estadios deportivos
10	Luces artificiales de anuncios
11	Luz diurna, amanecer o crepúsculo
12	Luz diurna, Sol muy cubierto
13	Luz diurna, Sol cubierto, sin sombras
14	Luz diurna, Sol algo cubierto, sombras suaves
15	Luz diurna, Sol, sombras nítidas
16	Luz diurna, pleno Sol

Figura 2.22 Cámaras. Valores EV característicos para varios tipos de escenas.





zona de luminosidad máxima o la intensidad de una zona de intensidad mínima o media.

Las *constantes de calibración* se determinan por análisis estadísticos a partir del análisis de varias fotografías con diferentes niveles de exposición y el juicio de un gran número de sujetos y cada fabricante los varía según su criterio. Para los valores de reflectancia, la ISO 2720:1974 recomienda un rango para *K* de 10,6 a 13,4. Canon, Nikon y Sekonic utilizan un valor de *K* en torno a 12,5 (muchos autores utilizan un valor de  $K = 12,7$ ) y Kenko y Pentax, en torno a 14. La diferencia entre estos valores supone en torno a 1/6 EV. Los valores de *K* están relacionados con valores de reflectancia media. Para los valores de incidencia, la ISO 2720:1974 recomienda un rango para *C* que varía según el tipo de fotómetro. Para un medidor plano (que varía con el coseno) los valores recomendados van de 240 a 400 si bien el valor más corriente es 250. Para un medidor hemiesférico (que varía según una cardioide), que suele dar resultados más precisos, los valores recomendados van de 320 a 540 si bien los valores más utilizados son 320 (p. ej., Minolta) o 340 (Sekonic).

El rango dinámico de una cámara puede expresarse, por tanto, de modo directo, por el número de pasos EV con los que cuenta. Las cámaras compactas de gama baja tienen (2014) un rango de unos 10 EV, las de gama media, de unos 12 EV y las SLR de gama alta, de unos 14 EV. Pero es preciso distinguir entre valores nominales y valores reales. Los valores nominales vienen dados por la precisión del conversor A/D de la cámara. La intensidad de la luz captada deberá traducirse a valores digitales que es lo que hace precisamente este conversor (su posición en la cadena de procesos varía según las cámaras, lo que complica más ciertos cálculos; en general suele estar situado tras las lecturas del sensor pero puede estar más atrás, después de los ajustes para el balance de blancos). Pero, como hemos visto, estos valores pueden venir afectados por diversas causas, entre ellas el ruido de fondo que limita el rango efectivo.

La tabla de la figura 2.23 proporciona las equivalencias entre la precisión del conver-

sor, en bits, y los rangos nominales resultantes. La mayoría de las cámaras digitales actuales cuentan con sensores y conversores A/D con capacidades que van de los 10 a los 14 bits. Sin embargo, el rango dinámico real está limitado por el ruido, que suele resultar más visible en las sombras. En la práctica es bastante menor, de los 5 a 9 pasos EV. El rango dinámico real de una buena SLR digital no suele ser mayor que 500:1.

No debe perderse de vista que las imágenes captadas por imágenes digitales se almacenan generalmente en un formato TIFF o JPEG, con 24 bits por píxel u 8 bpc (bits por canal), lo que supone 256 valores para cada canal. Pero esto supone una reducción, operada por los microprocesadores de la cámara, con respecto a lo que capta el sensor de una buena cámara digital. Si se guarda la imagen en formato RAW, la capacidad corriente es de 12 bits por canal, lo que proporciona 4096 valores discretos.

Por otro lado, si una imagen de 12 o 14 bits se abre en Photoshop, se convierte en una imagen de 16 bits. Como una imagen de 16 bpp permite codificar 65.536 valores ( $2^{16}$ ), sería de esperar que nos encontráramos con una lista disponible de 0 a 65.535. Pero no es así, pues Photoshop trata este tipo de formatos de modo diferente, como un “unsigned 15 bit plus 1”. El número de valores disponibles será, por tanto, 32.769 ( $2^{15} + 1$ ) y la lista disponible, de 0 a 32.768. Pero esto no es una codificación estándar. Volveremos a este tema más adelante, con los formatos HDR.

Hay otros factores a los que también volveremos más adelante. Aunque los CCD están diseñados para tener una respuesta lineal, los circuitos analógicos pueden introducir respuestas no lineales. Y esto es así debido a que, entre otras cosas, los fabricantes buscan que la salida sea más agradable para la gran mayoría de los usuarios aunque esto introduzca distorsiones prematuras. Si se conoce la función de proyección es posible recuperar esta transformación.

Veremos también, en el breve apartado sobre densitometría, que la curva característica de una película analógica tiene una forma en S con un tramo principal recto y dos extre-





mos, “el pie” y “el hombro”, curvados. En estos extremos, incrementos regulares de luminosidad se traducen en incrementos menores de densidad; pero la película sigue captando información. Los sensores digitales no tienen una respuesta tan fina. Sus fotositos se van llenando hasta que no les cabe nada más. La capacidad de un sensor, expresada en bits, mide la capacidad nominal de precisión aunque esto no quiere decir gran cosa, pues no implica que pueda captar una gama mayor sino solo más detalles dentro de la gama.

**Escáneres.** Los escáneres son bastante más antiguos que las cámaras digitales y han evolucionado mucho en los últimos años. Los primeros escáneres que se construyeron fueron los inventados por Hardy y Wurzburg y por Murray y Morse (1941). El primero fue desarrollado inicialmente para la Interchemical Corporation y, posteriormente, para la Radio Corporation of America. Puede considerarse, con muchas reservas, como el antecesor de los actuales *flatbed scanners* en la medida en que también efectuaba lecturas sobre un original dispuesto sobre un plano horizontal, por medio de un aparato de registro que iba recorriendo línea por línea este original y transmitiendo la señal

codificada a otro dispositivo. El segundo fue desarrollado inicialmente para la Eastman Kodak Company y, después, para la Time Incorporated y su filial Printing Development Incorporated (PDI), que dio su nombre a este primer escáner. Puede considerarse como el antecesor de los actuales escáneres de tambor, *drum scanners*. Los primeros escáneres estaban destinados a la grabación de clichés tipográficos por lo que cualquier comparación con los actuales puede resultar equívoca. Hasta la década de 1970 no se pudieron desarrollar la mayoría de los dispositivos actuales, entre los cuales tiene un papel fundamental la exploración por medio de rayos láser. Con todo, el sistema de registro óptico y conversión en señal electrónica de la señal óptica era similar.

El funcionamiento básico de los escáneres no es demasiado diferente al de las cámaras digitales, pues también se basan en sensores que captan la luz y la almacenan digitalmente por medio de conversores. Los escáneres actuales se pueden clasificar por su “número de bits”. La mayoría de los escáneres actuales son de 36 bits, aunque hay algunos de 30 bits y cada vez más de 42 o 48 bits. El número de bits está relacionado con el rango dinámico

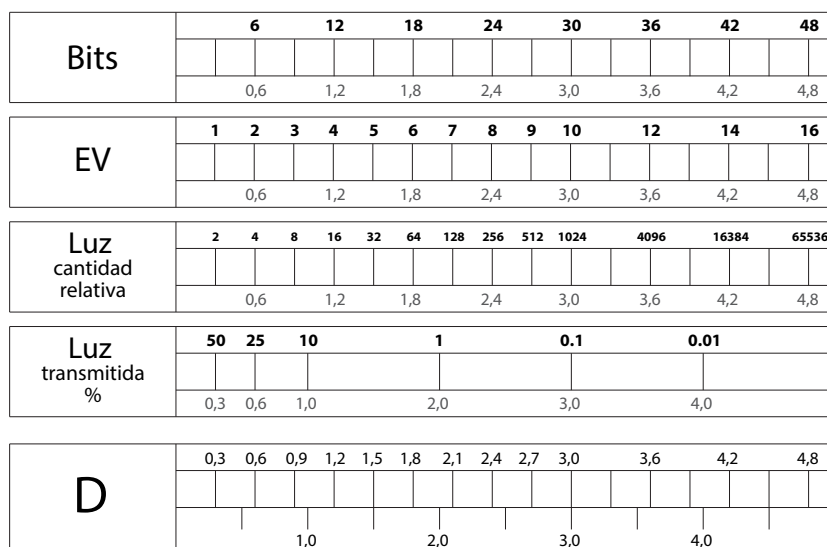


Figura 2.23 Relación nominal entre valores de densidad, luz transmitida, luz relativa, pasos EV y número de bits del conversor.



del escáner, pero la relación no es directa y deben tenerse en cuenta las características de los escáneres para interpretar adecuadamente estas cifras. Un escáner de menos bits que otro puede discriminar mejor las zonas oscuras debido a otros factores que cuentan más que el rango dinámico nominal.

Los escáneres profesionales captan la densidad de una transparencia por lo que las unidades con que trabajan son muy similares a los de la fotografía tradicional donde el concepto principal es la densidad. Opacidad, transparencia y densidad son nociones equivalentes. La *opacidad* se define como el cociente entre la intensidad de la luz incidente y la intensidad de la luz transmitida. El valor de la opacidad es siempre mayor que 1, por lo que resulta más cómoda de manejo que su inversa, la *transparencia*. Pero los valores resultantes en cualquiera de estos casos son muy altos, por lo que resulta más práctico utilizar otra noción equivalente, la *densidad*, que se define técnicamente como el logaritmo decimal de la opacidad o, lo que viene a ser lo mismo, como el logaritmo entre la luz incidente y la luz transmitida. Es decir:

$$D = \log(I/I_T)$$

Esta definición tiene la importante ventaja, aparte de ser más cómoda en la práctica, de que, como veremos más adelante, permite traducir a relaciones lineales las escalas perceptuales que se basan en series geométricas, no aritméticas.

Para ilustrarlo con un ejemplo, supongamos una transparencia sobre la que incida un rayo de luz de 1.000 unidades. Y que la densidad del negativo es tal que en las zonas más claras (más transparentes) se transmite el 95 % y en las zonas oscuras (menos transparentes) el 5 %. Las densidades correspondientes serían

$$D1 = \log(1.000/950) = \log(1,05) = 0,02$$

$$D2 = \log(1.000/50) = \log(20) = 1,30$$

Razonando de modo similar podríamos elaborar una tabla que nos relacione porcentajes de transmisión con transparencias. Una tabla así se muestra en la figura 2.25. Los valores

que se indican son valores extremos que rara vez se alcanzan en la práctica. Una densidad de 4,0 es un valor teórico, correspondiente a un negro completamente opaco, que no puede conseguirse en casos reales.

También podemos relacionar densidades con valores de exposición, lo que se correspondería con la cantidad relativa de luz transmitida. Y relacionar esto, a su vez, con la capacidad de un escáner en bits. La tabla de la figura 2.25 da la correspondencia teórica (suponiendo que no hay ruido y que las operaciones son perfectas) entre escáneres, bits y rangos dinámicos expresados en densidades.

El rango dinámico de un escáner es la relación entre la señal más brillante y la señal más oscura que puede detectar. Las dificultades principales vienen del lado oscuro y la capacidad de un escáner se relaciona sobre todo con su capacidad de discriminar las mayores densidades. Esto es debido a que todos los escáneres generan ruido. Y al amplificar la señal por medio de dispositivos especiales conectados a los sensores, se amplifica tanto la señal como el ruido. Esto, a su vez, implica que hay un mínimo no supe-

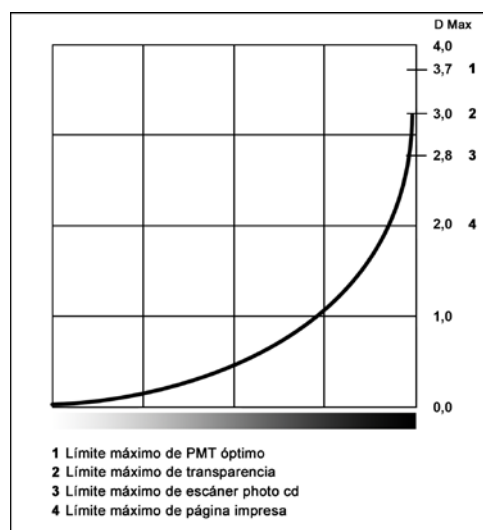


Figura 2.24 Escáneres. Valores de densidad característicos (adaptado de Kodak, 1995).



Tipo de escáner (max bits)	Valores posibles por canal (2 bits)	Densidad máxima ( $\log_{10}$ de v)
24 (8 x 3)	256	2,4
30 (10 x 3)	1.024	3,0
36 (12 x 3)	4.096	3,6
42 (14 x 3)	16.384	4,2
48 (16 x 3)	65.536	4,8

Figura 2.25 Escáners. Tipos de escáners por su número de bits y número de valores y densidades correspondientes.

able que viene dado por el ruido generado por el propio escáner.

Los escáneres que generan menos ruido son los denominados PMT (*photomultiplier tube*). Son tubos de vacío que operan con voltajes de más de 1.000 W y que se enfrían mediante sistemas de refrigeración para reducir aún más el ruido. Son aparatos que pesan del orden de 80 kg y pueden costar más de 30.000 €.

La densidad de una imagen se puede medir con densitómetros ópticos cuyo rango va de 0 (blanco puro) a 4 (negro puro). Cuanto mayor sea la densidad menor será el brillo. Los valores mínimos y máximos se denominan, respectivamente, Dmin y Dmax, y el rango dinámico de un escáner es la diferencia entre estos dos valores. Así, si un escáner tiene un Dmin de 0,2 y un Dmax de 3,1 su rango dinámico será Dmax - Dmin = 2,9. Es importante tener presente cuál es el rango dinámico de lo que se vaya a escanear pues, obviamente, si este es menor, como ocurre en la mayoría de los casos, los problemas son de otro orden. En el apartado sobre impresoras se da una relación de rangos dinámicos de densidades de soportes corrientes.

En la práctica se da corrientemente el valor de Dmax en lugar de Dmax - Dmin pues se asume que el valor Dmin es cercano a 0. Véase la tabla de la figura 2.25.

### Rangos de dispositivos de salida. Monitores. Impresoras

Los principales dispositivos de salida son los monitores y las impresoras. Este apartado se

limita a describir las características principales de estos dos dispositivos.

**Monitores.** El primer dispositivo electrónico de salida fue el tubo de rayos catódicos o CRT (*cathode-ray tube*) inventado por el físico alemán Ferdinand Braun en 1897. A lo largo de unos cien años desde esa fecha, y a través de múltiples contribuciones y mejoras técnicas, se convirtió en una de las herramientas electrónicas más utilizadas de la historia reciente. Sin embargo, desde finales del siglo xx está siendo substituido progresivamente por las pantallas planas de cristal líquido, LCD (*liquid crystal display*).

El sistema utilizado por ambos dispositivos es bastante diferente. Un monitor CRT funciona a partir de la emisión de tres flujos de electrones desde un extremo del tubo, uno por cada uno de los primarios RGB. Este flujo es regulado por unas bobinas magnéticas que los desvían para enviarlos a su posición, y que atraviesan una rejilla metálica que se utiliza para ajustar esta posición y para que lleguen a activar una capa de diminutos puntos de fósforo cuya mayor o menor intensidad se traduce en un punto de color. Un monitor LCD funciona, en la mayoría de los casos, a partir de la emisión de un flujo uniforme de luz desde un panel posterior, lo que se consigue por medio de una red de tubos fluorescentes muy finos y láminas difusoras de la luz emitida por los tubos. Este flujo de luz uniformizado atraviesa varias capas, principalmente dos filtros de polarización, uno horizontal y otro vertical, y dos capas de electrodos transparentes alineadas con la capa principal. Junto a estas capas de electrodos hay también, del lado trasero, una capa TFT (*thin film transistor*, película fina de transistores), con tantos transistores como píxeles, que incorpora la informa-



ción principal y, del lado frontal, un filtro de color que traduce las intensidades a colores primarios. La capa principal incorpora moléculas de cristal líquido cuya orientación se modifica por el voltaje suministrado, lo que a su vez afecta a la intensidad de la luz que la atraviesa. La figura 2.26 muestra un esquema que sintetiza el funcionamiento de estos dos dispositivos.

Por lo que respecta al rango dinámico, hay varias diferencias importantes. La primera es que la intensidad máxima de un CRT está limitada por la cantidad de energía que puede admitir un punto de fósforo. En el caso de un LCD, la intensidad está limitada por la intensidad de los tubos fluorescentes (que se puede aumentar aún más si se substituyen los tubos fluorescentes por LED). Sin embargo, desde el punto de vista de las intensidades mínimas, el CRT tiene la ventaja de que puede anularse totalmente la energía que llega a un punto mientras, que en el caso del LCD habría que bloquear por completo la luz que llega del panel posterior, lo que no se puede conseguir por completo. Así, ambos rangos dinámicos son limitados pero por razones diferentes: los CRT están limitados por su extremo superior, las intensidades máximas que puede alcan-

zar; los LCD están limitados principalmente por su extremo inferior, por las intensidades mínimas que pueden lograr.

El rango dinámico o factor de contraste es un parámetro que también se utiliza en los anuncios de monitores y aparatos de televisión. Pero los números que se dan en estos casos resultan equívocos pues se refieren a valores nominales que solo tendrían sentido si se correspondieran con situaciones reales, cosa que jamás ocurre. En los folletos o anuncios de los fabricantes, el contraste se da como la relación entre la luminancia mayor que pueden alcanzar las áreas brillantes de la pantalla y la luminancia menor que pueden alcanzar las áreas negras. En estos casos, es importante distinguir entre contraste estático y contraste dinámico. El contraste estático es el contraste que se alcanza simultáneamente en determinadas condiciones de observación. El contraste dinámico se refiere a los valores extremos que pueden alcanzarse variando las condiciones. Es un valor que resulta obviamente mayor que el estático.

Por otro lado, los aparatos de televisión modernos incluyen un procesador interno que analiza los valores medios de luminancia y

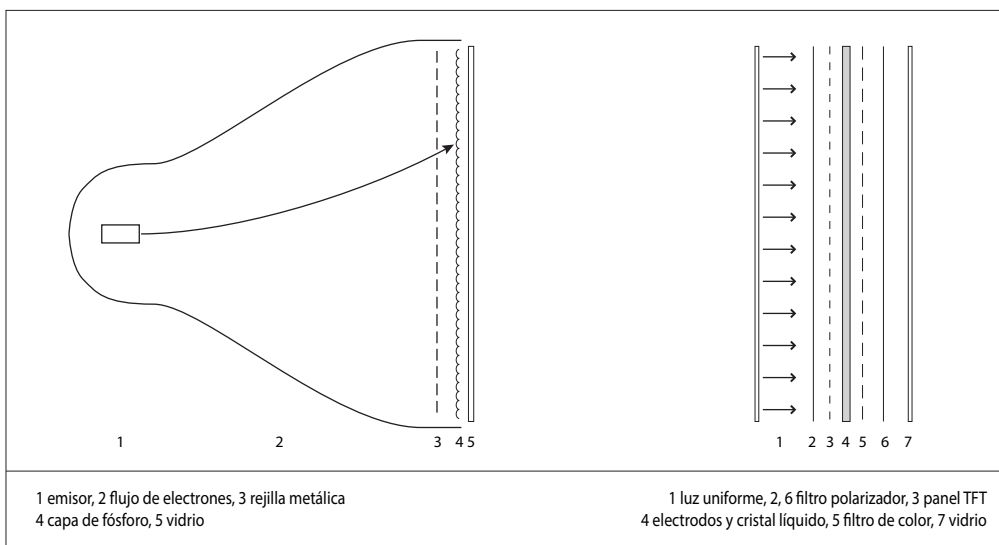


Figura 2.26 Monitores. Esquema del funcionamiento de un monitor CRT y un LCD.



ajusta la luminosidad de fondo para adecuarla a los diferentes tipos de escena. Los valores de contraste se refieren a los rangos alcanzados para diferentes tipos de escena, pero que nunca se alcanzan en una escena concreta.

No es realista hablar de los “rangos dinámicos” anunciados, pues corresponden a situaciones que no se pueden dar en la realidad por lo que hablaré más bien de “rangos efectivos”. Supongamos que tenemos un televisor de un gran rango dinámico, con un contraste nominal de 3.000:1 que correspondería a luminancias extremas de 300 cd/m<sup>2</sup> y 0,1cd/m<sup>2</sup>. Y que estamos contemplando una escena en la que determinadas zonas están al máximo y otras al mínimo, algo que, por otro lado, casi nunca ocurrirá. Y supongamos también, que hay una luz tenue en la sala en que estamos que envía al monitor 5 cd/m<sup>2</sup>. En este caso, las luminancias serían 300 + 5 y 0,1 + 5. La relación de contraste sería entonces 305/5,1 que es aproximadamente 60:1, es decir, un valor mucho más bajo que el anunciado.

Los monitores CRT tienen un rango efectivo que varía entre los 50:1 y los 200:1. Los monitores planos, TFT, más o menos corrientes tienen un rango de unos 300:1. Los monitores de gama más alta, como los Eizo, pueden alcanzar valores del orden de los 500:1. Los proyectores de cine tienen factores de contraste del orden de 1.000:1. Hay monitores de alto rango dinámico, a los que me volveré a referir más adelante, como el Brightside, que alcanzan rangos nominales de 15.000:1, para luminancias extremas de 3.000 cd/m<sup>2</sup> y 0,2 cd/m<sup>2</sup>, cifras que, para que tengan algún sentido requieren formatos de alto rango dinámico, como veremos.

Además, hay que tener en cuenta también que se utilizan diferentes métodos de medir el contraste (principalmente por la técnica *full*

*on/of*, que mide la relación entre extremos, y por la técnica *Ansi*, que mide la relación con respecto a la luminosidad media), lo que aún complica más las cosas.

Resumiendo: en general puede decirse que cuando nos encontramos con la especificación de un factor de contraste superior a 500:1, este valor se ha obtenido bajando al máximo el punto negro, hasta extremos incompatibles con el punto blanco al máximo, es decir, en condiciones incompatibles con condiciones corrientes de observación.

*Impresión sobre papel.* En este caso, el rango dinámico es muy limitado pues depende directamente de la reflectancia. Los mejores papeles fotográficos tienen una reflectancia del orden de un 90 %. Y si se utilizan pigmentos negros cubrientes se pueden alcanzar valores de un 2 %. Estos dos valores son los que determinan el rango y son independientes de la iluminación. Si observamos una fotografía impresa con ese rango bajo una luz no muy intensa, pongamos que de 200 luxes, tendremos unas intensidades extremas de 180 (200 x 0.9) y 4 (200 x 0.02) lo que supone un rango de 1:45 (180/4). Y si la intensidad es mayor, pongamos que 800 luxes, tendremos 720 (800 x 0.9) y 16 (200 x 0.02) lo que supone igualmente un rango de 1:45 (720/16).

Si nos limitamos a las impresoras de chorro de tinta, que abarcan el mayor segmento del mercado, con tres fabricantes principales (Epson, Canon y HP) puede decirse que hay dos tipos: las que utilizan colorantes (*dyes*) y las que utilizan pigmentos. Los colorantes se decoloran con el tiempo con bastante rapidez. Si partimos de la base de que queremos que la impresión dure un tiempo suficiente en buen estado, la segunda opción es, en principio, preferible. Sin embargo, es más fácil adaptar los colorantes al mecanismo de propulsión de minúsculas gotas

<i>Tipo de monitor</i>	<i>Rango de luminancias</i>	<i>Factor de contraste</i>
CRT	0,50 a 100	100:1
LCD	0,40 a 120	300:1
LCD gama alta	0,25 a 120	500:1
Plasma gama alta	0,15 a 120	800:1

Figura 2.27 Monitores. Rangos característicos.

sobre el papel de las impresoras de chorro de tinta. Pero también ha habido avances notables que permiten utilizar partículas de pigmento más pequeñas. Y, por otro lado, los fabricantes de colorantes han encontrado modos de hacer los colorantes más estables. También hay que decir que muchas impresoras de gama alta no incluyen una única tinta negra sino varias, a menudo 4, con el valor más denso denominado, según los casos *mate* o *fotográfico*.

Esto quiere decir que para asegurarse buenos resultados habrá que conseguir información actualizada sobre los dos extremos que nos importan: el tipo de papel y el tipo de pigmento negro utilizado por la impresora que vayamos a utilizar.

En cualquier caso, está claro que estos valores de reflectancia no pueden competir con los rangos reales ni con los de los otros dispositivos que hemos visto. Esto obliga a plantearse otras consideraciones.

Por estas mismas razones los pintores siempre han intentado realzar las zonas más brillantes, sea recubriéndolas de láminas de oro, sea depositando gruesas capas de pinturas del pigmento blanco más opaco de que podían disponer (blanco de plomo). Pero, sobre todo, recurriendo a efectos de contraste para crear la ilusión de que las zonas luminosas eran más luminosas de lo que realmente eran. Las obras de De la Tour o Rembrandt son ejemplos magistrales de cómo utilizar estos recursos.

En el caso de simulaciones digitales, la alternativa equivalente para una imagen que se va a imprimir será redistribuir adecuadamente los valores por medio, entre otras cosas, de las técnicas de proyección de tonos (*tone mapping*) que veremos más adelante.

**Impresión de diapositivas.** Si el medio es transparente en lugar de reflectante, como ocurre con los negativos o las diapositivas, el rango dinámico aumenta. Por otro lado, las condiciones en que se contemplan, con el entorno a oscuras, favorecen el contraste y aumentan aún más el rango dinámico.

Sin embargo, debido a que, para controlar la transparencia, las diapositivas dependen de la densidad del medio que se deposita sobre el soporte, lo que obliga a pasar por un proceso tedioso de revelado que cada vez se utiliza menos y que, por otro lado, dependen de una iluminación que se proyecta a través de ellas, que es algo que ya hacen los proyectores ligados a medios digitales, han caído en desuso, por lo que no merece la pena detenerse demasiado en describir sus características.

Los negativos de buena calidad pueden tener un rango dinámico de unos 2,8 D. Y las diapositivas del orden de 3,2 D. Sin embargo la diapositiva tiene una curva característica más inclinada (un factor gamma mayor) y en los extremos la densidad es mayor, a diferencia de lo que ocurre con los negativos, en los que la imagen está invertida y el ruido se desplaza a las zonas más claras en donde es menos apreciable. Por estas razones, el rango dinámico efectivo de una diapositiva es de 5 a 6 EV mientras que el de un negativo puede ser de 9 o 10 EV.

### Construcción de escalas en dispositivos analógicos. La densitometría

La siguiente pregunta que hay que hacerse es cómo organizar concretamente estos rangos dinámicos. En el capítulo anterior ya hemos visto cómo se construirían, en general, escalas de grises. Y cómo la relación entre los valores

<i>Tipo de soporte</i>	<i>Rango de reflectancias</i>	<i>Factor de contraste</i>
Papel de periódico	8 a 80	10:1
Papel brillante	4 a 85	20:1
Papel fotográfico, gama alta	1 a 90	90:1
Diapositiva	(no aplicable)	100:1
Negativo, gama alta	(no aplicable)	1.000:1

Figura 2.28 Impresión. Tipos y rangos característicos.



de luminosidad o reflectancia, emitidos por un soporte determinado y los valores de brillo, percibidos por nuestro sistema visual, es una relación no lineal, de tal modo que a un valor de brillo medio (50 %) le corresponde un valor de reflectancia de 18 %, aproximadamente. Sin embargo, la distribución general, sobre todo en los extremos de la escala, no es trivial. La construcción material de una escala de grises implica, por consiguiente, el estudio del modo en que puede controlarse la respuesta específica de un soporte real a la acción de la luz.

Por otro lado ¿cómo traducir esta relación a características concretas del soporte? La respuesta científica a esta pregunta se desarrolló, en primer lugar, en el ámbito de la fotografía mediante métodos que podrían extrapolarse a otros ámbitos. La *sensitometría* es la ciencia específica de donde provienen la mayoría de los conceptos que se han extrapolado desde el campo de los soportes analógicos al campo de los soportes digitales. Se denominó así al estudio científico del comportamiento de los materiales fotográficos bajo la luz. Estos estudios se originaron a partir de los trabajos de Ferdinand Hurter y Vero Charles Driffield, hacia 1876, razón por la cual las curvas que presentaron por primera vez se denominan a menudo “curvas H-D”, una denominación que aún se sigue utilizando. Dado que la parte principal del estudio

la constituye el análisis de la variación de densidad del material, posteriormente se ha utilizado el término *densitometría*, que podemos considerar como prácticamente sinónimo pero que tiene mayor sentido para diferenciarlo de otros estudios similares que no se basan en el análisis de la densidad de transparencias o impresiones sobre papel. Este es el término que usaré aquí.

En el apartado sobre dispositivos artificiales, al resumir las características de los escáneres, ya he introducido el concepto de densidad, *D*, equivalente a opacidad y transparencia y que se define como el logaritmo de la opacidad.

Si una muestra de una emulsión fotográfica se expone a una luz de características fijas, los granos de plata se irán haciendo más densos, lo que se traducirá posteriormente en un oscurecimiento de la imagen, en una mayor opacidad. Esta opacidad o densidad se puede medir con un aparato especial, un densitómetro, y los resultados se pueden llevar a un gráfico cuyas abscisas representan los logaritmos de las luminancias y cuyas ordenadas representan las densidades. De este modo se obtiene la denominada *curva característica* de la emulsión.

El ennegrecimiento o la densidad dependen de tres factores principales: el tipo de emulsión, el tipo de revelado y la exposición luminosa a que se someta la película. Si se

Dispositivo	Factor de contraste	Rango D	Rango EV	Bits
Impresión, papel corriente	10:1	1,0	3,3	10
Impresión, papel de calidad	40:1	1,6	5,3	16
Monitor CRT	50:1	1,7	5,6	
Impresión fotográfica	100:1	2,0	6,6	20
Formato imagen 8 bpp	256:1	2,4	8,0	24
Monitor LCD	300:1	2,5	8,2	
Monitor gama alta	500:1	2,7	9,0	
Cámara digital simple	1.000:1	3,0	10,0	30
Escáner simple	1.000:1	3,0	10,0	30
Formato imagen 12 bpp	2.048:1	3,3	11,0	
Cámara digital media	4.000:1	3,6	12,0	36
Escáner medio	4.000:1	3,6	12,0	36
Cámara SLR alta	16.000:1	3,0	14,0	42
Escáner gama alta	25.000:1	4,4	14,6	
Formato imagen 16 bpp	65.536:1	4,8	16,0	48

Figura 2.29 Resumen de rangos dinámicos de dispositivos y soportes digitales.





mantienen constantes los dos primeros factores y se va variando sistemáticamente la exposición, se irán obteniendo unos valores de densidad que quedarán directamente relacionados con los valores de exposición. De este modo, será posible obtener, para cada tipo de emulsión y revelado, una referencia objetiva que ayude a controlar la reproducción. Este procedimiento puede ampliarse a cualquier sistema, incluidos los sistemas informáticos, que necesiten una herramienta objetiva de control de las luminancias de una imagen.

La figura 2.30 muestra una curva característica donde los valores representados corresponden, sobre el eje de las X, al logaritmo relativo de las luminancias para intensidades de iluminación que pueden variar entre 1 y 1.000 cd y que, en casos especiales, pueden llegar a 10.000. Estos valores dan un rango de 0 a 3 puntos, correspondiente a las densidades (logaritmo de las opacidades).

Esta curva característica presenta cuatro zonas prototípicas:

- a) La zona de velo. Es un pequeño trazo horizontal al principio de la curva. Este trazo indica que la película no es totalmente transparente y que los puntos más claros, que han recibido poca o nula exposición, han quedado velados, debido a fenómenos secundarios que tienen lugar durante el revelado.
- b) La zona de subexposición. Es un trazo curvo que, junto con el anterior, forma lo que se denomina el “pie” de la curva. En esta zona no hay una correspondencia lineal entre aumento de exposición y aumento de densidad. Es la zona a la que corresponden las zonas más oscuras, las sombras que, se dice, “caen” sobre el pie.
- c) La zona de exposición. La zona principal es la que corresponde al tramo recto central. Es la zona sobre la cual caen principalmente los tonos bajos, medios y altos. En esta zona, un aumento constante de las exposiciones da lugar a un aumento constante de las densidades. El rendimiento, sin embargo, puede variar en función de la pendiente de este tramo. Si

su inclinación es de  $45^\circ$  habrá una correspondencia perfecta entre exposiciones y densidades. Si la inclinación es superior a  $45^\circ$ , esto querrá decir que un aumento de las exposiciones genera un aumento proporcional, pero relativamente inferior, de densidades. Si la inclinación es inferior a  $45^\circ$ , esto querrá decir que un aumento de las exposiciones genera un aumento proporcional, pero relativamente superior, de densidades.

- d) La zona de sobreexposición. Es un trazo curvo al final de la curva que forma lo que se denomina el “hombro”. Al igual que en la de subexposición, se pierde la correspondencia lineal entre exposiciones y densidades. Es la zona a la que corresponderán las zonas más claras, las luces que, se dice, “caen” sobre el hombro. Por añadidura, si se continúa la exposición, hay una inversión del efecto previsible, fenómeno que se denomina “solarización” y que se traduce en una disminución de las densidades al aumentar la exposición.

La inclinación de la curva en la zona de exposición, el tramo recto, se mide por la tangente del ángulo de inclinación, es decir, por el cociente entre el incremento de densidad y el incremento del logaritmo de la exposición. Este valor se denomina *gamma* y como veremos está relacionado, principalmente por razones históricas y teóricas, pues no es exactamente lo mismo, con el factor

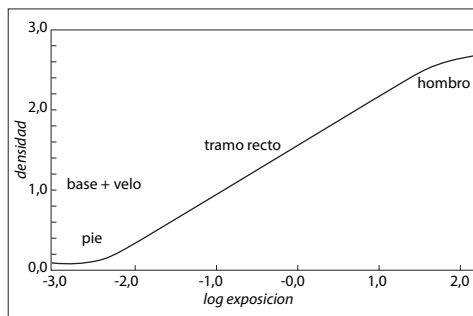


Figura 2.30 Curva característica de emulsiones fotográficas.

*gamma* de corrección de distribución de intensidades en dispositivos digitales. Es decir:

$$\gamma = \Delta D / \Delta \log \text{exposición}$$

El parámetro *gamma*, en sensitometría, es una medida del *contraste*. Otra medida del contraste se obtiene por medio del *gradiente medio*. Se denomina así a la pendiente de una línea que conecta dos puntos concretos de la curva que pueden estar situados más allá del tramo recto.

Los rangos dinámicos equivalentes son fáciles de calcular si trabajamos con logaritmos, ya los hemos visto en los apartados anteriores. Un rango de densidad 2,0 equivale a un factor de contraste de 100:1 y uno de 3,0 a uno de 1.000:1. Un negativo analógico corriente tiene un rango de densidades del orden de 1.8 (según Kodak), que equivale a un factor de contraste de 60:1. Este rango puede situarse con facilidad sobre el rango máximo que se utiliza en las curvas características, que es de 3,0 y permite jugar con desplazamientos en ambos sentidos, en pasos de 0,3 de log E (equivalentes a un *f-stop* o un EV) para aclarar u oscurecer el resultado. El arte del revelado puede entenderse así como el arte de desplazar un rango dinámico limitado, captado en un caso concreto, sobre un rango más amplio dado por la curva característica de la película.

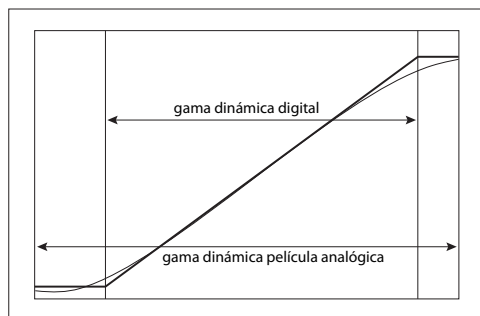


Figura 2.31 Curva característica de emulsiones fotográficas comparada con digitales.

Las películas digitales tienen un comportamiento diferente a las analógicas por muchas razones. Una de ellas es la diferente respuesta en los extremos. Aparte del hecho de que la gama dinámica de una película analógica es más amplia que la de una digital, su comportamiento en los extremos es distinto pues la pendiente recta, similar a la de la digital, se curva suavemente, lo que implica que se adapta a las variaciones, de tal modo que un aumento o disminución de la exposición se traduce en cambios menores en la densidad. Esto es importante pues quiere decir que se puede retener más detalle en los extremos. Sin embargo, en las digitales esto no ocurre: los fotodiodos no cuentan con este efecto amortiguador, se saturan bruscamente, como puede apreciarse en la figura 2.31, que muestra una curva características de una imagen digital superpuesta a una analógica. Esto requerirá, como veremos, recurrir a métodos especiales para salvaguardar detalle en los extremos.

### Construcción de escalas en dispositivos virtuales. El factor *gamma*

La medida del contraste, y su relación con el parámetro *gamma*, que nació en los estudios de sensitometría para poder expresar las propiedades de una emulsión, se ha ido desarrollando a partir de la aparición de dispositivos digitales o, más exactamente, dispositivos analógicos, pero que reciben información digital (pues un monitor, un escáner o una impresora son objetos reales).

La construcción de escalas en dispositivos virtuales implica dos problemas que trataremos por separado. En primer lugar, la adopción de una curva adecuada que permita redistribuir unos valores que, inicialmente, son lineales y que no se corresponden con el modo en que el sistema visual percibe intervalos iguales. En segundo lugar, la especificación de un número suficiente de valores para que los degradados aparezcan como perfectamente continuos. Como veremos,



ninguno de estos dos temas tiene una solución perfecta. Por esta razón conviene conocerlos bien, pues será necesario recurrir a técnicas especiales para compensar estas soluciones imperfectas.

Para no recargar la exposición, lo que sigue se basará en el análisis de solo uno de los dos dispositivos principales de entrada, las cámaras digitales, y solo uno de los dos de salida, los monitores.

**La curva de redistribución y el factor gamma.** El primer problema es, por tanto, dicho de un modo escueto, cómo relacionar intensidades con brillos. La intensidad se expresa en unidades tales como vatios y es una medida lineal, es decir, que aumenta de modo directamente proporcional al aumento en estas unidades. Los sensores que captan la luz en cámaras digitales acumulan fotones de un modo lineal antes de transferirlos a un conversor A/D. Los cálculos que se efectúan en un programa de simulación obtienen resultados que se computan inicialmente por medio de intensidades. Así que, tanto si partimos de imágenes derivadas del mundo real como de imágenes sintéticas, generadas internamente en un ordenador, nos encontramos con distribuciones inicialmente lineales.

El correlato fotométrico de la intensidad es la luminancia: la intensidad de radiación ponderada por una curva de distribución espectral adecuada al sistema visual humano, tal como hemos visto en el capítulo 1. La luminancia puede obtenerse, para tres colores primarios, mediante una suma ponderada de los valores de los tres primarios.

El brillo es un atributo perceptual que no se corresponde con ninguna medida objetiva. Nuestra estimación del brillo se basa en experimentos y estadísticas que confirman que su distribución no es lineal y que una luminancia de un 18 %, aproximadamente, (para valores normalizados tales que la máxima luminancia presente sea 100 %) nos parece corresponder a un brillo medio. Para proporcionar un correlato objetivo al brillo, la CIE propuso el concepto de *luminosidad* (*lightness*), deno-

tado por  $L^*$ , como un correlato objetivo del brillo. Y lo definió mediante una fórmula que ya he citado en el capítulo 1 y que puede considerarse como una variante de la raíz cúbica de la luminancia:

$$L^* = 116(Y/Y_n)^{1/3} - 16; \quad 0,008856 < (Y/Y_n)$$

donde  $Y_n$  es la luminancia del color blanco de referencia. Si se normaliza la luminancia al blanco de referencia no es necesario computar el cociente. Para valores muy bajos, cercanos al negro, tales que el cociente  $Y/Y_n$  sea inferior a 0,008856, la curva resultante se substituye por un tramo lineal. Si en esta fórmula se hace el cociente  $Y/Y_n = 0,18$  el valor resultante de  $L^*$  será aproximadamente 50.

Como también hemos visto, la capacidad de discriminación de valores tiene un umbral que es del orden de 0,01. Para la fórmula de la CIE esto se concreta en el valor denominado  $dL^*$  o *delta L-star*, que es igual a 1 para el rango normalizado de 0 a 100.

La propia fórmula, con estas correcciones, y las variantes que hemos visto en el capítulo anterior, ya nos indican que la solución es un compromiso pues no es posible encontrar una fórmula realmente objetiva. De ahí que se encuentren diversas variantes en los dispositivos digitales. Todos usan alguna función potencia con un exponente variable. Este exponente variable se denomina *gamma* y está relacionado con la pendiente y la curvatura de la curva. Dado que afecta a la distribución de valores y al contraste, se relaciona con el parámetro del mismo nombre que hemos visto en el apartado anterior sobre densitometría.

Hay una cierta confusión que también conviene aclarar sobre las funciones potencia, exponencial y logarítmica, que son similares pero que no son lo mismo. La función potencia tiene la forma general  $x^a$ . Sus variantes para diferentes valores de  $a$  (*gamma* en nuestro caso) se muestran en la figura 2.32. La función exponencial tiene la forma general  $a^x$ . Y la función logarítmica, la forma general  $\log(x)$ . Las funciones exponenciales y logarítmicas son inversas la una de la otra (pues  $a^{\log_a x} = x$

y  $\log_a(a^x) = x$ ). Pero la función potencia no es lo mismo que estas dos funciones aunque su forma sea similar para determinados valores.

Las cámaras digitales llevan incorporados programas que transforman las intensidades lineales en distribuciones dadas por funciones de potencia que simulan la respuesta del sistema visual humano. Otro tanto ocurre con los programas informáticos, que trabajan internamente con intensidades lineales pero que deben redistribuirlas para mostrarlas en pantalla.

Estas transformaciones se denominan *corrección de gamma*. La figura 2.33 muestra una curva de corrección de gamma característica de una cámara de vídeo. Como se puede apreciar en la figura, la curva tiene en este caso un exponente gamma de 0,45 (1/2,2) con un segmento recto en la parte inferior para minimizar el ruido de los sensores. Esto es un ajuste ideal que no se lleva a cabo en los monitores corrientes.

El valor de 0,45 (1/2,2) no es el valor que se obtendría de una redistribución teórica. El valor inicial es 0,40 (1/2,5). Pero este valor está reducido porque los fabricantes de televisores decidieron que las circunstancias más corrientes serían que los usuarios vieran

el televisor en un ambiente oscuro pero con algo de luz ambiental, lo que reduce el contraste. Los fabricantes de televisores asumen un factor de contraste en las salas de estar de sus usuarios de 30:1 aproximadamente. Para compensarlo, aumentaron el contraste adoptando el valor citado, que acabó por ser incorporado a la codificación interna de las propias cámaras de rodaje.

Por otro lado, los monitores tienen incorporada una función de transferencia basada en un determinado valor *gamma*. Esta función de transferencia, denominada *curva de reproducción de tono* (TRC, *Tone Reproduction Curve*) es no lineal y es similar a la no linealidad de la curva perceptual humana. Es importante tener en cuenta que los controles del monitor pueden alterar esta curva, tal como se muestra en la figura 2.34, que muestra en el eje horizontal valores de voltaje y en el eje vertical luminancias resultantes. La curva y los valores (adaptados de Poynton) corresponden a un monitor CRT y se efectúan sobre el flujo electrostático, no sobre los fósforos. Aunque los monitores LCD tienen una tecnología diferente que invalida las fórmulas utilizadas para un CRT, suelen incorporar funciones internas que replican la tecnología de los CRT mediante algoritmos de transferencia de valores.

El menú del cualquier monitor ofrece dos controles principales para ajustar la salida. El control de *contraste*, un término más bien equívoco, ajusta la intensidad global de la pantalla. El control de *brillo* ajusta el desplazamiento de los tonos. Para ajustarlo, un método sencillo es escoger una imagen con una zona que se sabe que es negra y ajustar el brillo hasta que esta zona se vea completamente negra.

Así que la situación, en general, es la que se muestra en la figura 2.35. La primera salida, tanto si se capta la imagen con una cámara, como si se genera internamente, es lineal. Esto debe redistribuirse por medio de funciones internas, incorporadas a la propia cámara o al programa de generación de la imagen, que se basan en una función poten-

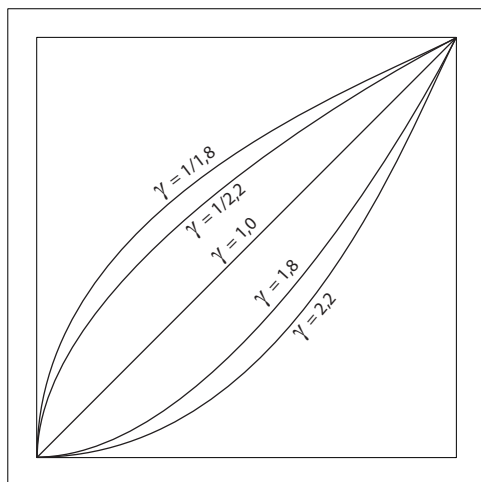


Figura 2.32 Funciones potencia  $xy$  para diferentes valores de  $\gamma$ .



cia con una gamma que, por lo general, es de 0,45(1/2,2) en el caso de los PC. Silicon utiliza una gamma distinta, de 0,68 (1/1,47) y Macintosh otra distinta, de 0,56 (1/1,8). El resultado se almacena en un archivo, en el caso de una cámara, o en el *frame buffer*, una memoria temporal previa a su salida por pantalla, en el caso de un ordenador (lo que también puede guardarse en un archivo). En el caso de un ordenador, la redistribución se aplica a los datos almacenados en el *frame buffer* por medio de una *lookup table*, una tabla que almacena los valores y sus equivalentes RGB corregidos por la función de transferencia. Una tabla que a veces puede ser accesible por el usuario y a veces no, y que puede estar documentada o no. La salida final se reajusta por la curva de ajuste del propio monitor. Dado que esta es la única fase a la cual el usuario tiene acceso corrientemente, la denominación "corrección gamma" se aplica por lo general a esta fase final, aunque no hay una sino varias funciones de transferencia de corrección implicadas.

Todo esto introduce una indeterminación considerable que, entre otras cosas, dificulta la transferencia de resultados entre diferentes aplicaciones.

**Número de valores.** La segunda pregunta que nos hacíamos al comienzo era ¿cuántos valores se necesitan para simular un degradado continuo sin que aparezcan discontinuidades? La respuesta inicial viene dada del valor de umbral de 0,01 (1 en un rango de 100:1). Esto quiere decir que si comenzamos por el extremo inferior, con una determinada luminancia mínima, necesitamos mantener la relación  $I_n = I_{n-1} \times 1.001$ .

Desarrollando esta relación resulta que  $I_n = I_0 r^n$ . Si la intensidad final es 100 tendremos que  $1,01^n = 100$ . Con lo que  $n = \log(100)/\log(1,01)$ , es decir que  $n = 2/\log(1,01)$  lo que da un resultado de 460 valores para una codificación basada en este factor, no lineal. Esto requiere 9 bits (512 valores). Si repitiéramos un razonamiento similar para una codificación lineal encontraríamos que son necesarios 9.900 valores, 14 bits (13 bits nos darían 8.192 y 14 bits 16.384).

Estos 9 bits son 1 bit más de lo que se utiliza corrientemente: 8 bits (256 valores). Sin embargo, debido a varios factores, entre ellos que, como hemos dicho, las condiciones de observación siempre son muy inferiores a los valores óptimos, este es el valor utilizado desde hace muchos años, entre otras cosas porque se adapta mejor al modo en que se empaquetan los bits (en paquetes de 8 bits = 1 byte). Si bien, como veremos en la sección siguiente, la creciente exigencia de los usuarios avanzados está llevando a modificar estos límites por medio de formatos de alto rango dinámico.

Así que podemos aceptar el valor de 256 valores como primera respuesta, algo reba-

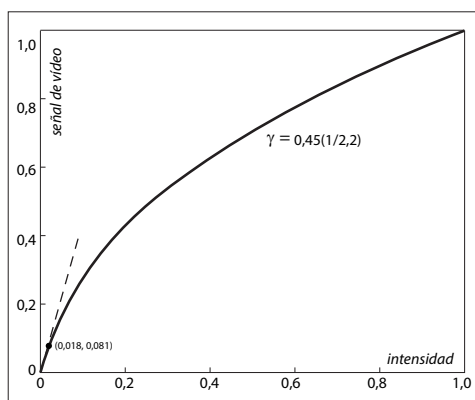


Figura 2.33 Corrección de gamma con un valor de 0,45 (1/2,2) para una cámara de video.

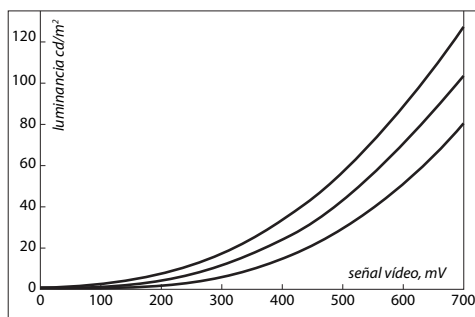


Figura 2.34 Curvas de corrección de tono de un monitor CRT. Las diferentes curvas corresponden a diferentes regulaciones de brillo y contraste de los controles del monitor.

jada, a la pregunta inicial. Pero esto es solo la primera parte de la respuesta. La segunda parte es ¿cómo se distribuyen estos valores? Pues la distribución es automática y debido a que la entrada inicial es lineal hay un problema importante: esta distribución inicial hace que la información que necesitamos (la distribución de luz en la escena) se distribuya de un modo inadecuado para las transformaciones no lineales que siguen.

Para explicar el problema con un ejemplo sencillo: si equiparamos el rango dinámico con una determinada altura que tenemos que salvar por medio de una escalera, el número de valores que tenemos sería el número de peldaños de esta escalera. Pero esta es una escalera muy rara en la que los escalones no tienen la misma altura: los primeros son muy altos y los últimos, muy bajos. Más exactamente, nosotros la percibimos como regular pero realmente está construida de este extraño modo para que nosotros podamos percibirla como regular. Esto da lugar a problemas internos que es necesario entender.

Los sensores de las cámaras, como ya he dicho, son lineales. Esto quiere decir que los valores que almacenan se distribuyen tal como se muestra en la figura 2.36, arriba a la izquierda. En esta figura se muestra, en la

parte superior, una distribución de 8 valores (3 bits), para que se pueda visualizar más claramente el problema y, en la parte inferior, fragmentos iniciales y finales de una distribución corriente de 256 valores (8 bits).

Como puede apreciarse en la parte superior, una distribución lineal hace corresponder un valor medio con una intensidad media. Sin embargo, nuestro sistema visual no funciona así, como ya hemos visto: lo que percibimos como intensidad media corresponde a una intensidad de un 18 % aproximadamente. Para que los valores almacenados por los sensores se adapten a la distribución perceptual humana, *en algún momento* habrá que introducir una curva de corrección como la que se muestra en la parte superior derecha. Pero esto hace que el reparto se distorsione: el estiramiento de los bits inferiores hace que toda la información que contienen, que es muy inferior a la de los bits superiores, se reparta en la zona de valores oscuros, inferiores al gris medio. Y a la inversa: la compresión de los valores superiores hace que la información quede reducida a un espacio que puede resultar insuficiente.

En el párrafo anterior, se ha destacado la frase “en algún momento” porque esta conversión puede tener lugar en la propia cámara

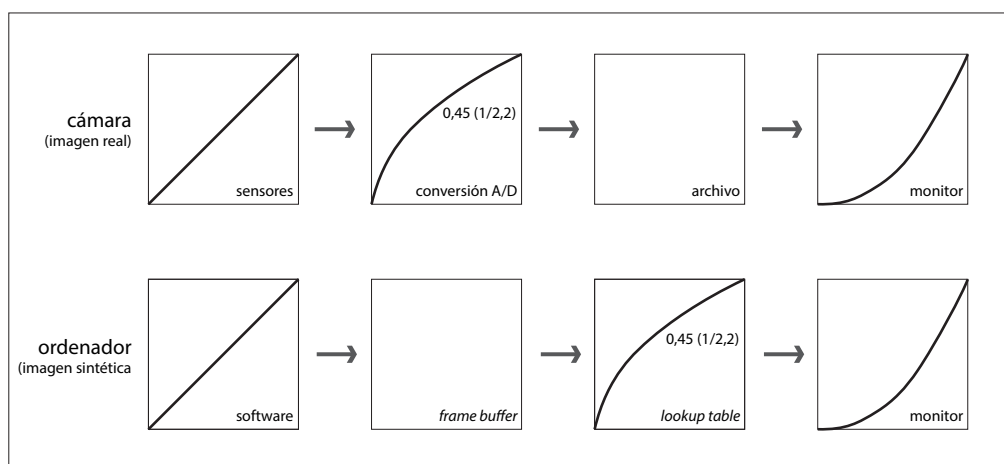


Figura 2.35 Cadena de conversiones en el caso de una cámara digital y de un ordenador (PC) hasta llegar a la salida por el monitor.



o el procesamiento posterior, lo que complica bastante las cosas. Todos los fabricantes de cámaras digitales incorporan a los conversores A/D sistemas de corrección que aseguren que los usuarios verán los resultados ya corregidos.

También debe tenerse en cuenta, si se quieren realizar comprobaciones o análisis, en programas como Photoshop, que los valores dados por el tercer parámetro del espacio cromático HSB no son fiables, pues a veces se obtienen a partir de una ponderación de los tres componentes, lo que no es coherente. Es más recomendable cambiar al espacio cromático Lab y tomar el valor dado por el parámetro  $L$  que es independiente y estable.

**Impresión.** El análisis de lo que ocurre cuando la salida viene dada por una impresora es más complicado, pues implica de un modo prácticamente inevitable el uso de métodos de gestión de color. Véase el apartado correspondiente en el libro sobre *simulación de materiales* citado en la *introducción*. Se deben tener en cuenta al menos las observaciones siguientes relativas solo a los valores acromáticos.

Aparte de los desplazamientos cromáticos, que también pueden afectar a los tonos neutros, es importante tener presente que los tonos medios se imprimirán más oscuros que lo que se ve en pantalla. Esto es debido a un fenómeno denominado ganancia de punto (*dot gain*) que es debido, por un lado, a la dispersión de las tintas y, por otro, a efectos ópticos internos. Para un tono medio, la ganancia de punto puede ser de algo más de un 20 %. Esto quiere decir que, en impresión tradicional por semitonos, si un área determinada estaba cubierta en un 50 % por pequeños puntos, el cubrimiento real puede ser mayor del 70 %. En impresión digital, los fenómenos básicos son algo diferentes pero el resultado es similar: los tonos tienden a oscurecerse. Algunos programas, como Photoshop, incluyen controles para compensar esta diferencia. Pero dada la multiplicidad de factores implicados, principalmente el tipo de papel y de tinta, es difícil generalizar.

Estos desplazamientos de valor implican, de nuevo, la utilización de curvas de salida de corrección de tonos. Los profesionales de la edición impresa utilizan curvas pregrabadas para corregir los tonos según el tipo de papel e impresora y redistribuir los valores para compensar estos desplazamientos. Pero, de nuevo, esto entra dentro de un tema complejo, la gestión del color por medio de perfiles cromáticos personalizados, que no podemos abordar aquí. Me remito al apartado mencionado anteriormente o a la información general que se puede encontrar sobre este tema.

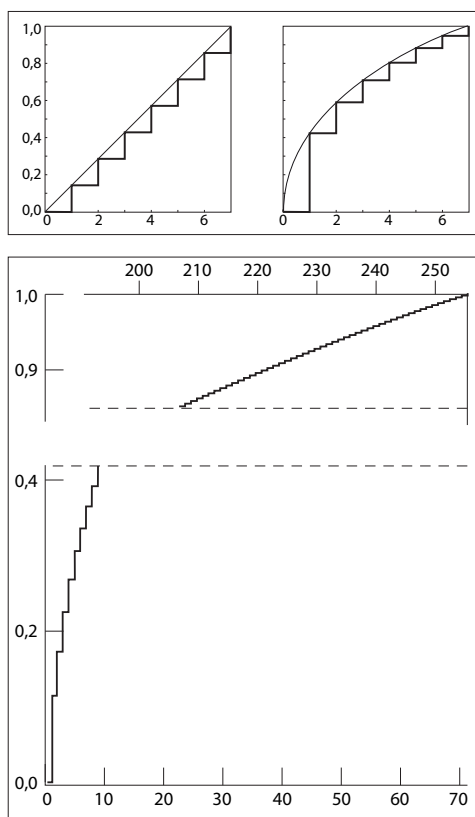


Figura 2.36 Efecto de la redistribución automática de una información lineal por medio de una curva de redistribución gamma. En la parte superior se muestra un ejemplo simplificado a 8 valores (3 bits). En la parte inferior, el detalle de las zonas altas y bajas, para un ejemplo corriente de 256 valores (8 bits).





## Formatos de alto rango dinámico (HDR). Introducción. Conceptos básicos y aplicaciones

Las limitaciones de los formatos de 8 bits por canal, con un máximo de solo 256 valores, han llevado a desarrollar un formato alternativo con capacidad de almacenar un número mucho más alto de valores que permita, idealmente, reproducir adecuadamente el rango dinámico muy superior de las escenas reales. Este tipo de formatos se desarrolló a finales de la década de 1990 y su uso se ha extendido cada vez más en los últimos años. En este apartado doy una introducción general a las ideas y los desarrollos teóricos. En los capítulos siguientes se explicarán técnicas y aplicaciones concretas.

Una imagen HDR (*high dynamic range*, alto rango dinámico) o HDRI (*high dynamic range imaging*) es una imagen que almacena suficiente contraste y definición tanto en las zonas claras como en las oscuras. Esto no es posible si se utiliza cualquier método de captación de imágenes convencional pues en estos casos es preciso ajustar la exposición a la luminosidad media. Como sabe cualquier aficionado a la fotografía si, por ejemplo, se saca una fotografía a pleno día con luz brillante y se ajusta la exposición para que las zonas iluminadas queden bien expuestas, los detalles de las zonas en sombra se fundirán en una zona oscura homogénea. Y si, a la inversa, se saca una fotografía de un interior y se ajusta la exposición para que los objetos del interior queden bien expuestos, los objetos que aparezcan a través de una ventana aparecerán quemados, sobreexpuestos, fundidos en una zona clara homogénea.

Sin necesidad de sacar fotografías, el efecto es similar al de mirar a un interior desde un exterior (nos parece uniformemente negro) o un exterior desde un interior (nos parece uniformemente blanco). Pero si nos concentramos en una zona evitando la luz lateral, en ambos casos, la vista se adapta lentamente hasta que reconocemos detalles que antes eran invisibles.

Una imagen HDR permite la fusión de estas diferentes condiciones de adaptación en un mismo archivo: el detalle se conservará en ambos extremos del rango. La información así obtenida no se puede visualizar en un dispositivo corriente pues su rango dinámico es muy inferior.

Los archivos HDR se usan de dos modos principales: como fuente de iluminación de escenas 3D por medio de técnicas que se verán más adelante y como base para redistribuir los tonos de diversos modos y crear a partir de ellos imágenes LDR (*low dynamic range*), sea seleccionando parte del rango, sea combinando diferentes partes de este rango. El resultado, en este último caso, es una imagen que tiene un aspecto singular, sorprendente y más bien artificioso, y que puede utilizarse con finalidades artísticas o prácticas en diversos tipos de aplicaciones que se resumirán más adelante.

Los antecedentes de esta idea se remontan a los años 1930 y 1940 y a las fotografías tomadas por Charles Wickoff de explosiones nucleares grabadas con diferentes exposiciones en una película especial de tres capas. Pero las primeras contribuciones digitales se deben a Greg Ward Larson, un nombre bien conocido por sus numerosas aportaciones a métodos de simulación, que volveremos a encontrar y que, entre otras cosas, presentó en 1985 el programa *Radiance*, que utilizaba unidades fotométricas reales para los cálculos de iluminación y que codificaba la información obtenida mediante valores abiertos, en lugar de números enteros, en un rango limitado de 0 a 255 (8 bits) como era, y es, corriente.

En 1997 y 1998, cuando la industria había desarrollado técnicas diversas que permitían sacar mayor partido de la idea de Ward, Paul Debevec presentó dos ponencias en SIGGRAPH (véase Debevec 1997, 1998) que marcarían el inicio de una nueva fase y la popularización de estos métodos. La primera ponencia de Debevec era un resumen de su tesis doctoral "Recovering high dynamic range radiance maps from photographs". El método desarrollado por Debevec y Malik,



entre otros, se basa en obtener varias imágenes del mismo motivo pero con diferentes exposiciones. Estas tomas se combinan de tal modo que la imagen resultante muestra detalles tanto en las zonas brillantes como en las oscuras.

Uno de los problemas obvios que aparecen a continuación, como ya he dicho, es como mostrar estas imágenes en dispositivos que no son HDR sino LDR. La respuesta a estos problemas es de dos tipos. Por un lado, se han desarrollado métodos para convertir imágenes HDR en imágenes LDR sin perder excesivo detalle por medio de una redistribución adecuada de los tonos. Como esta es la solución actual, pero no es única y conviene conocer sus peculiaridades, la teoría se abordará en el siguiente apartado y la práctica en los capítulos posteriores correspondientes.

Por otro lado (para quien no tenga problemas de presupuesto) se han fabricado monitores especiales, a los que ya me he referido y que permiten, hasta cierto punto, visualizar directamente imágenes grabadas en formatos HDR. Esto va unido a desarrollos técnicos incorporados a las tarjetas gráficas más avanzadas. DirectX, por ejemplo, ha desarrollado extensiones que soportan el software Shader Model, que permite la representación de imágenes HDRI.

Las imágenes HDRI se pueden *obtener* de varias formas. La primera y más sencilla es adquirirlas en bibliotecas, páginas web o empresas especializadas que proporcionen imágenes en este formato. El inconveniente de esta vía es que habrá que pagar por ellas y que no serán originales. La segunda es tomar varias fotografías de un entorno real, con diferentes valores de exposición, y combinar luego las imágenes obtenidas por medio de programas especiales. La tercera es generar una imagen adecuada en un programa de simulación, con un sistema de iluminación global y grabar el resultado en formato HDR. La cuarta es obtenerlas directamente mediante dispositivos especiales (y muy caros). En los capítulos siguientes se explicarán más ampliamente todos estos métodos.

Los formatos HDR se utilizan para un número creciente de *aplicaciones*. Las principales son las siguientes: a) Fotografía digital. En este caso, pueden acabar por ser una alternativa importante al formato RAW, con mayor capacidad de procesamiento que los formatos corrientes, pues su capacidad de almacenar información y luego darle diversos tipos de salida es tan extraordinaria que, aparte de los métodos citados, es previsible que aparezcan cámaras con mayor capacidad para generar directamente archivos HDR (por ejemplo, aumentando el rango de *bracketing*, obtención automática de subexposiciones y sobreexposiciones en un rango amplio); b) Animación digital. Lo dicho con respecto a la fotografía digital se aplica igualmente a las animaciones. Estas aplicaciones ya se han iniciado pues el formato HDRI se utilizó por primera vez en algunas escenas de la película *X-Men the Movie* (2001) y posteriormente se ha utilizado en muchas más como *Spiderman 2*, *Batman Begins* o *X-Men 2*. Y el formato OpenEXR se ha utilizado en películas como *Harry Potter and the Sorcerer's Stone* (2001), *Men in Black II*, *Gangs of New York* y muchas más; c) Edición digital. Los programas de edición de imagen más conocidos, como Photoshop, ya incorporan los formatos HDR y algunas herramientas para actuar sobre este tipo de formatos. Esto incrementa también el tipo de tratamientos que se pueden aplicar a imágenes digitales; d) Simulación visual físicamente correcta. Si se lleva a cabo una simulación con luces fotométricas, con sistemas de iluminación global los resultados pueden ser mucho más precisos que lo que los formatos y monitores corrientes pueden mostrar; e) Realidad virtual. Lo dicho con respecto a la simulación visual vale para la realidad virtual, esto es, para las simulaciones visuales interactivas. La utilización de formatos HDR permitirá crear escenarios que incorporen mayor información y más precisa; f) *Image-Based Lighting* (IBL). Una aplicación importante y que veremos más adelante es utilizar imágenes HDR como “mapas de irradiación” para simular la iluminación de una escena en programas de simulación.



Un mapa de este tipo se puede utilizar para simular la iluminación de un espacio real por medio de los valores de radiancia almacenados en el mapa, lo que permite obtener resultados equivalentes a los del espacio real; g) *Remote sensing*. Las imágenes captadas por satélites y por dispositivos especiales almacenan mucha más información de la que el ojo humano puede distinguir y de la que un monitor corriente puede mostrar. Guardar estos resultados en formato HDR posibilita que esta información pueda ser recuperada para todo tipo de usos.

Como decía al comienzo, las imágenes HDR utilizan *formatos* especiales. Una imagen grabada en formato HDR guarda la información como valores en coma flotante. Esto quiere decir que podría guardar información tal que representara un color blanco por un valor como 12.345,67. Si este valor se guardara en un formato corriente, que solo admite 256 valores dados por números enteros, el valor se multiplicaría por un factor para reducirlo al rango disponible y los decimales se truncarían, lo que equivaldría a recortar la escala de luminosidades.

Ya hemos visto que una escena más o menos corriente abarca al menos una gamma de 10.000:1 y en exteriores, 100.000:1. Con 8 bpp tenemos una gamma muy pobre, de poco más de 100:1 (256:1). Con 16 bpp llegamos a más de 32.000 (pues son 15 bits efectivos en algunos programas), lo que será más que suficiente para la gran mayoría de los casos. Con 32 bits tendríamos, en principio, más de 4.290 millones (4.294.901.760) o  $\sim 4,3 \times 10^9$ .

Hay que recordar que las imágenes HDR no necesitan corrección gamma pues pueden obtenerse valores más finos en cualquier punto, con lo que los datos de la imagen pueden permanecer en un espacio lineal que se redistribuirá por medio de *tone mapping* para poder ser visualizado en un dispositivo LDR.

La primera codificación HDR que se utilizó fue *Radiance*, un formato desarrollado por Greg Ward hacia 1987 para su sistema de simulación del mismo nombre y que dio lugar a nuevos sistemas de *rendering*, los denominados *physically based renderers* pues,

a diferencia de todos los demás programas de *rendering*, calculaban la imagen final con valores reales de radiancia obtenidos, a su vez, de datos de radiancia sobre la escena físicamente correctos. Esto requería trabajar con valores precisos, en coma flotante. Y esto es lo que llevó a Ward a desarrollar un formato especial, que inicialmente utilizaba la extensión PIC. Pero tuvo que cambiarla pues ya se utilizaba en otros formatos, y finalmente se adoptó la propuesta por Paul Debevec, con la extensión HDR. El formato HDR se presenta en dos modos: RGB y XYZE (tres primarios imaginarios, XYZ y un exponente, E). En los dos casos, los bits se organizan de modo que hay una mantisa para cada color y un exponente común (8 + 8 + 8 + 8). Es decir, que los tres primeros canales almacenan la información de color primario pero el cuarto canal introduce un exponente que amplía los valores de luminancia. De hecho, es un simple truco matemático por el que el valor de cada píxel se obtiene según la fórmula  $((R,G,B)/255) \times 2^{(E-128)}$ . Supongamos por ejemplo que un píxel tiene una especificación RGBE de (100, 150, 200, 150), es decir, que corresponde a un azul claro (100,150,200) bastante luminoso. Su valor resultante será:  $((100, 150, 200)/255) \times 2^{22} = (1.644.825, 2.467.237, 3.289.650)$ . El rango dinámico resultante de utilizar 32 bits de esta forma es equivalente a utilizar 96 bits, lo que equivaldría a 76 órdenes de magnitud o más de 250 EV, mucho más de lo que podemos percibir, pues ya hemos visto que los escenarios reales no suelen sobrepasar los 20 EV en condiciones extremas ni los 16 bits en condiciones corrientes.

Otra codificación de alta precisión y bastante utilizada es *LogLuv*, también desarrollado por Ward y utilizado por Silicon Graphics. En este caso, la luminancia se codifica por separado y los parámetros UV se corresponden a la codificación del sistema de la CIE LUV, una codificación desarrollada por la Commission International de l'Eclairage en 1976, como alternativa a la codificación CIE XYZ de 1931, para proporcionar una distribución espacial más acorde con las diferencias de per-



cepción de intervalos cromáticos del sistema de visión humano. Sin embargo, los programadores no lo han utilizado mucho debido a que no usa directamente los primarios RGB, lo que implica complicaciones adicionales. Esta codificación está incorporada a una variante de una versión del formato TIFF, (TIFF 32 o TIFF en punto flotante), que incluye una codificación RGB de 32 bits (con 8 bits para U, V y 16 para la luminancia), con una enorme gama dinámica, de 79 órdenes de magnitud, lo que permite una gran precisión y se utiliza a menudo para conversiones entre diferentes aplicaciones. La variante original, sin embargo conseguía, mediante unos recursos sumamente ingeniosos, codificar toda la información principal en tan solo 24 bits.

La principal alternativa al HDR de Radiance es *OpenEXR*. Este formato fue desarrollado por Florian Kainz para ILM (Industrial Light and Magic) la empresa promovida por George Lucas, hacia 1999, para crear muchos de los efectos especiales que aparecen en sus películas (la primera película rodada íntegramente con EXR fue *Harry Potter and the Sorcerer's Stone*). En 2002 se liberó, presentándose como código abierto. Hay dos versiones del formato, la versión "half", de 16 bits, y la versión de 32 bits. La versión "half", o "half RGB", más utilizada, usa un formato de números de 1 bit para el signo, 5 bits para el exponente y 10 bits para la mantisa o valor significativo. Esta mantisa permite guardar  $2^{10} = 1.024$  valores por cada canal de color. Proporciona un rango dinámico equivalente a 30 pasos EV. Su precisión es mayor que la de Radiance y se comprime muy bien.

En la práctica se utilizan formatos HDR de 16 o 32 bits. Por ejemplo, cuando abrimos un archivo de tipo HDR desde un programa como 3ds Max nos encontraremos con tres opciones: *real pixels* (32 bpp), *16 bits per channel, linear* (48 bpp) y *8 bits per channel, linear* (24 bpp) ¿Qué quiere decir todo esto? Tanto HDR como OpenEXR utilizan imágenes de 32 bits, lo que supone valores en coma flotante. Sin embargo, un factor clave del éxito de OpenEXR es que también admite mapas

de 16 bits (semi-coma flotante). Esto permite obtener una calidad suficiente para la mayoría de los casos sin la carga de memoria que supone trabajar con formatos de 32 bits por canal. Piénsese que un archivo de 16 mp de resolución (por ejemplo, de unos 3.872 x 2.592 px) que ocupa unos 2,5 mb en formato JPG, ocupará unos 19 mb en formato TIF (LogLuv), unos 28 mb en formato HDR, unos 58 mb en formato EXR y unos 116 en formato PFM (*portable float map*, véase más adelante).

Hay otros formatos que merecen ser mencionados. En 2002, Kodak introdujo el ERI-JPEG (*extended range imaging JPEG*) con la idea de utilizar los metadatos Exif (*exchangeable image file format*, etiquetas de datos incrustadas en los archivos que comenzaron a utilizarse extensamente desde 1998) de los JPEG para añadir información adicional que permitiera ampliar el rango dinámico. Sin embargo, hasta la fecha, estos datos solo son interpretados adecuadamente por programas de tratamiento de imágenes de Kodak, con lo que el problema de fondo acaba siendo similar al del gran número de formatos RAW que están a la espera de algún tipo de codificación universal. Otro intento interesante fue el formato FJPEG (*floating point JPEG*), desarrollado por el profesor alemán Helmut Dersch (que también ha hecho notables contribuciones al desarrollo de sistemas de Panorama Stitching con su suite Pano Tools) hacia 2003. La idea era similar: utilizar los metadatos para ampliar el rango dinámico. Pero el sistema utilizado por Dersch se parece más al utilizado por Ward para Radiance. Básicamente, se crea un canal adicional que representa la imagen en escala de grises. El problema en este caso, según parece, es que la aplicación PTViewer, también desarrollada por Dersch en código abierto, es la única que lee este tipo de archivos y no parece haber, por ahora, gran interés por parte de otras empresas en utilizarlo. Una tercera mención es el formato JPEG-HDR, también desarrollado por Greg Ward para Disney hacia 2004 y que también utilizaba, como los dos anteriores, los metadatos del JPEG corriente. En este



caso, para almacenar una versión LDR, elaborada con *tone-mapping*, del HDR original. Como en las anteriores, la dificultad está en conseguir que aplicaciones corrientes sean capaces de leer y trabajar con estos datos. Por esto, hay más probabilidades de éxito en los formatos en coma flotante que ya están desarrollando varias compañías (Windows o Adobe, entre otras), aunque probablemente nos volveremos a encontrar con el problema de las codificaciones cerradas. Si menciono estas alternativas es porque muestran las posibilidades y el interés de ampliar los formatos corrientes para incorporar rangos dinámicos ampliados. En cualquier caso, los dos formatos más utilizados en la actualidad (2014) siguen siendo Radiance (.hdr) y OpenEXR (.exr).

Una razón adicional para grabar imágenes en formato HDR o EXR es que hay más probabilidades de que estos datos sean legibles en un futuro próximo que si se guardan en un formato RAW ligado a una cámara específica. Actualmente, muchos fotógrafos aficionados (y por descontado los profesionales) prefieren guardar sus fotografías en formato RAW para ser ellos quienes apliquen las correcciones necesarias, en lugar de los sistemas automáticos que se aplican al guardarlas en un formato corriente como JPEG y que son destructivos de los datos originales. Pero cada fabricante tiene su propio formato RAW (.arw, .sfr, .sr2 para Sony, crw, cr2 para Canon, .nef para Nikon, .orf para Olympus, .raf para Fuji, .raw, .rw2 para Panasonic, etc.). La mayoría proporcionan 10, 12 o 14 bits, lo que da bas-

tante margen para el procesamiento posterior y para conseguir rangos dinámicos medios. Sin embargo, no hay un estándar único a pesar de que han surgido algunas iniciativas importantes. Una es el formato DNG (*digital negative*), introducido por Adobe hacia 2004. Pero no deja de ser un formato ligado a una determinada compañía comercial, pese a que su uso se ha extendido considerablemente, por lo que no se le puede calificar de formato estándar.

Debido a estos problemas, un grupo de fotógrafos formó, a partir de 2005 aproximadamente, la OpenRAW.org ([www.openraw.org](http://www.openraw.org)) con la finalidad de promover el uso de formatos abiertos, lo que dio lugar, en 2006, a un informe notable sobre el estado de la cuestión, aunque aún no se ha conseguido convencer a las empresas de que adopten un estándar compartido. Otra alternativa es el formato DCRAW, una codificación abierta, introducida por Dave Coffin por esas mismas fechas, que mantiene una base de datos que almacena varios tipos de cámaras y los traduce a este formato. Pero como suele ocurrir con estas iniciativas generosas, su continuidad es imprevisible. Por esta razón, como decía, es una buena idea, por ahora, grabar las imágenes que realmente nos interesen en formato hdr o exr.

Otro formato que conviene tener presente es PFM (*portable float map*) o PBM (*portable bitmap map*, una versión idéntica utilizada, con esta extensión, por Photoshop). Un formato PFM contiene básicamente la misma información que contendría un archivo de 32 bits, con los valores de cada píxel en coma flotante. Su

Formato	Rango dinámico	Precisión	Bits/Píxel
Radiance RGBE	76,0	1,0 %	8R + 8G + 8B + 8E
Open EXR	9,6	0,1 %	16R + 16G + 16B
LogLuv 32 bits	38,0	0,3 %	$16\log L + 2 \times 8(u', v')$
LogLuv 24 bits	4,8	1,1 %	$10\log L + 14(u', v')$
TIFF 48 bits	5,4	1,0 %	$16\log R + 16\log G + 16\log B$
TIFF IEEE 96 bits	79,0		32R + 32G + 32B
TIFF Pixar (log TIFF)	3,8	0,4 %	11R + 11G + 11B

Figura 2.37 Principales formatos HDR.





principal ventaja es la simplicidad de la estructura, que se puede leer fácilmente y convertir a formatos corrientes de alto rango dinámico, pues contiene poco más que una cabecera en la que se especifican las dimensiones de la imagen seguida de los datos correspondientes a cada punto. Cualquier programador con conocimientos elementales puede traducir esta información a un formato HDR.

También hay que recordar que existe una versión TIFF a veces denominada FP TIFF, TIFF32 o TIFF Float, muy similar al anterior, PFM, aunque su cabecera es más compleja e incluye datos sobre el espacio cromático, el tipo de compresión y algunas cosas más.

### Distribución de valores en formatos HDR. Tone mapping

Tal como decía al comienzo, el problema obvio con que nos encontramos en la práctica es que las imágenes HDR se visualizan corrientemente en dispositivos LDR. Por tanto, si se utilizan imágenes HDR será necesario comprimir el alto rango dinámico de estas imágenes en un rango mucho más limitado. Este procedimiento se denomina genéricamente “redistribución de tonos”, “proyección de tonos” o “mapeado de tonos”, una traducción no muy ortodoxa del término original inglés, *tone mapping*.

Esto se lleva a cabo por los operadores de proyección de tonos o TMO (*tone mapping operators*). Un TMO es básicamente un algoritmo que analiza la imagen y redistribuye los valores de un modo adecuado.

En general, estos operadores se dividen en dos categorías principales: operadores globales y operadores locales. Los primeros algoritmos de *tone mapping* se comenzaron a desarrollar en la década de 1990 por Greg Ward, Paul Debevec y Erik Reinhard, entre otros. Algunos de los resultados principales están recogidos en una obra de referencia publicada por Reinhard, Ward, Pattanaik y Debevec en 2005 (véase Referencias).

Las ideas principales se introdujeron por Tumblin y Rushmeier (en 1993) y ese mis-

mo año se publicó el primer algoritmo, debido a Chiu et al (en 1993) que era bastante limitado y daba lugar a gradientes inversos, halos característicos en torno a zonas brillantes o fuentes de luz. Larson et al. (en 1997) desarrollaron operadores globales mejorados con ajustes locales. Avances importantes posteriores fueron los desarrollados por Pattanaik et al. (1998) o Tumblin y Turk (en 1999) aunque los resultados eran demasiado artificiales. Los operadores más recientes, de Ashikhmin (2002), Fattal et al. (2002), Reinhard et al. (en 2002), Durand y Dorsey (2002) y Choudhury y Tumblin (en 2003) han resultado mucho más efectivos a la hora de separar diferencias de contraste que son significativas de las que no lo son.

Con todo, ninguna de estas técnicas ha sido validada por investigaciones psicofísicas por lo que sus resultados se pueden calificar de adecuados o inadecuados tan solo a partir del juicio personal de quien los use. De hecho, la mayoría de ellos incluyen parámetros que se dejan a disposición de los usuarios para que los utilicen según su buen saber y entender.

Casi todos comparten algunas características comunes tales como que los datos de entrada se calibran en valores reales, el color se trata de modo similar o que la mayoría usan versiones desenfocadas de la imagen de entrada (Reinhard, 2005, p. 224) y la compresión en general es logarítmica aunque en algunos casos es lineal.

La principal distinción, y la que utilizaremos más adelante, es la de operadores globales y locales. Sin embargo, Reinhard, en la obra citada (2005), menciona cuatro categorías principales que resumo a continuación. En capítulos posteriores desarrollaré ejemplos de aplicación de algunos de estos operadores.

a) Operadores globales. En principio, la misma curva se aplica a todos los píxeles de la imagen, pero dado que los monitores ya están ajustados para una respuesta no lineal el resultado es una imagen muy oscura que hay que corregir con otra curva de ajuste de *gamma*. Se pueden utilizar curvas loga-



rítmicas corrientes pero es preferible utilizar algoritmos más específicos. Hay dos algoritmos bastante utilizados, uno desarrollado por Drago en 2003, que básicamente extiende la curva logarítmica de respuesta (con una base que varía entre 2 y 10) para abarcar el rango más alto. Y otro desarrollado por Reinhard hacia 2002, que se basa en una adaptación del sistema de zonas de Ansel Adams y que comentaré con algo más de extensión más adelante.

b) Operadores locales. Parten de la idea de que el sistema visual humano se adapta, en general, a zonas determinadas, más que al conjunto de la escena. Los primeros en utilizar este criterio como punto de partida fueron Chiu *et al.* (1993), que se basaron también en la observación de que los artistas a menudo realzan el contraste en zonas de valores altos, oscureciendo lo que las rodea, para simular rangos mayores en el conjunto de la escena. Así, el nivel de adaptación se deriva para cada píxel en función de sus vecinos. El problema que surgió con estos primeros algoritmos es que aparecían halos indeseados. El algoritmo desarrollado posteriormente por Pattanaik *et al.* (2000) mejoraba los resultados mediante un sistema complejo que buscaba modelar todos los niveles conocidos propios del sistema visual humano. Otro algoritmo posterior desarrollado por Ashikhmin (2002) reducía la complejidad de este sistema aplicándolo tan solo a los aspectos relevantes para la compresión de la imagen, con lo que resultó bastante más rápido.

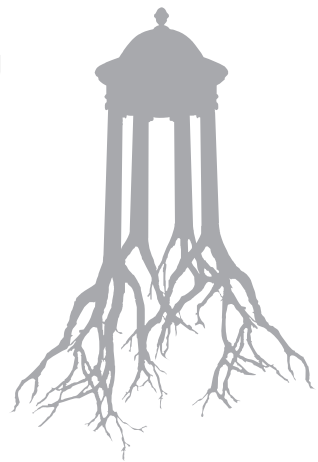
c) Operadores en el dominio de la frecuencia. La idea principal de la que se parte es que una imagen puede ser considerada como formada por un componente LDR con frecuencias altas y un componente HDR con frecuencias bajas. Si se pueden separar estos dos compontes bastará ajustar uno de ellos. Para ello, se reduce el rango dinámico de la imagen a partir de la frecuencia espacial de la región de imagen considerada. El algoritmo más utilizado es el desarrollado por Durand y Dorsey (2002), que se basa en un filtrado bilateral que desenfoca ligerament el conjunto de

la imagen pero preserva las altas frecuencias en los bordes.

d) Operadores en el dominio de los gradientes. Los gradientes de una imagen son más suaves para las frecuencias bajas. Estos gradientes pueden comprimirse con operadores especiales. El algoritmo más utilizado para aplicar esta idea es el desarrollado por Fattal *et al.* (2002) que comprime los gradientes en zonas de baja frecuencia de modo variable para preservar los detalles que más interesen.

Como ya he dicho, en los capítulos siguientes veremos métodos específicos para aplicar estas técnicas.





→ 3



# Sistemas de cálculo de iluminación

## 3.1 Introducción

### Procedimientos básicos y marco general del proceso de cálculo

Para crear un escenario virtual hay que seguir una serie de pasos que incluirán, en general, la especificación de algún tipo de emisor virtual de energía luminosa. El procedimiento general es similar, tanto si se utilizan métodos de iluminación local como métodos de iluminación global, y se resumen en los 7 pasos siguientes que se ilustran en la figura 3.1.

1. Modelado geométrico y organización de la escena.
2. Asignación de atributos materiales a las superficies.
3. Creación y ajuste de luces virtuales.
4. Preparación de un punto de vista y una cámara virtual.
5. Configuración del cálculo de la iluminación. En sistemas de iluminación local este paso es automático pues el sistema lo configura internamente. En sistemas de iluminación global requiere ajustar diversos parámetros, según el tipo de cálculo elegido, y llevar a cabo una serie de pruebas.
6. Representación (transformación de los valores obtenidos por el cálculo en colores de píxel). Esta transformación depende de operadores de tono y de la corrección de la distribución para adaptarla al sistema visual del observador. En sistemas de iluminación local estas correcciones son automáticas. En sistemas de iluminación global requieren ajustes complementarios por parte del usuario.
7. Obtención de una imagen que se grabará en un formato adecuado.

Esta descripción está referida al proceso de trabajo tal como es experimentado por el

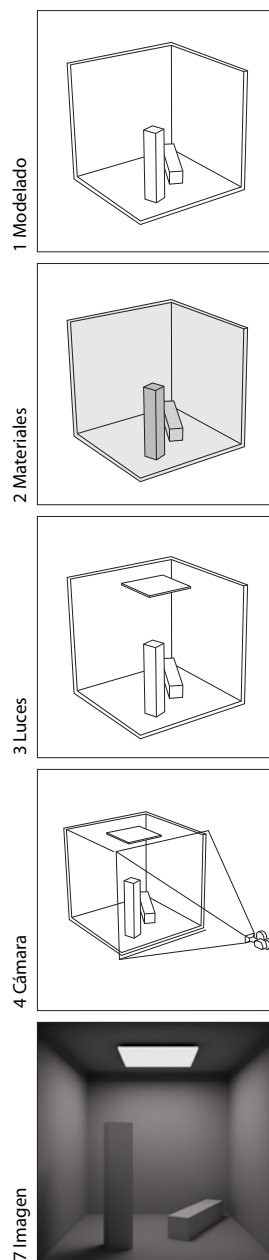


Figura 3.1 Marco general del proceso de cálculo.



usuario. Internamente, el proceso es mucho más complejo, afecta principalmente a los estadios 5 y 6 y conviene conocer alguna de sus peculiaridades, por lo que esta descripción se ampliará en el capítulo siguiente.

### Sistemas de iluminación local

Se denominan sistemas de iluminación local, por oposición a los sistemas de iluminación global que han surgido posteriormente, a los sistemas que se basan en luces estándar que, como ya hemos visto en el capítulo 2, se basan exclusivamente en la comparación entre la dirección del vector luz y la dirección del vector normal a la superficie considerada.

Como también se ha visto en el capítulo 2, los métodos locales de cálculo de iluminación tienen ventajas y desventajas evidentes: son sencillos y rápidos pero proporcionan imágenes muy simples, que no representan adecuadamente el modo en que la luz ilumina los objetos y los espacios. Esto no quiere decir que no sean útiles para un gran número de casos, y es muy importante calibrar bien qué es lo que se necesita, pues puede muy bien ocurrir que un cálculo local proporcione imágenes relativamente esquemáticas pero más adecuadas para la representación de un proyecto. Hay que tener en cuenta también la importancia creciente que ha adquirido en los últimos años lo que se conoce en la literatura técnica como representación no fotorrealista o NPR por sus siglas en inglés (*non-photorealistic rendering*), con nuevos sistemas de simulación de materiales similares a los de los dibujos o los cómics (el material *ink'n paint* de 3ds Max o los *toon materials* de otros programas).

Si se necesita, por las razones que sea, una simulación de la iluminación más exacta, las posibilidades son muy variadas y se necesita una comprensión mínima del estado del arte (en 2014) y de las peculiaridades técnicas de los diferentes métodos de Iluminación Global o GI, por sus siglas en inglés (*global illumination*). Este capítulo está dedicado a

describir la evolución histórica y los principios teóricos de estos sistemas.

## 3.2 Sistemas de iluminación global

### Resumen de la evolución histórica

Los métodos principales que existen en la actualidad han ido evolucionando en múltiples direcciones a partir de los dos métodos principales que surgieron en la década de 1980, *ray tracing* (introducido por Turner Whitted en 1980) y *radiosity* (introducido por Goral, Torrance, Greenberg y Battaile en 1984).

Una base teórica más firme para los algoritmos y métodos de GI vino con la publicación, por James Kajiya en 1986, de la *rendering equation*. En ese mismo artículo, Kajiya introducía una nueva técnica, *path tracing*, que tendría posteriormente una gran importancia, tanto teórica como práctica. A partir de estas fechas fueron apareciendo diversos métodos para aproximarse a una solución suficiente de esta ecuación, métodos que se engloban generalmente bajo la etiqueta general de métodos de Montecarlo. Esto dio lugar a diferentes técnicas, que se siguen utilizando con variantes propias de los programas comerciales que las incorporan.

En 1985, Greg Ward presentó un nuevo sistema de software (*Radiance*) y, en 1988 publicó un artículo, ampliado en 1992 (con Heckbert), en el cual proponía un método de recolección e interpolación que pasó a denominarse de radiancia o irradiancia (o *irradiance catching*). En 1995, Jensen dio a conocer otro método, *photon mapping*, que se amplió con diversas contribuciones en los años posteriores.

Durante la década de 1990 aparecieron diversos métodos que combinaban los sistemas principales por medio de “múltiples pasos” (*multipass methods*). Algunos de los sistemas más populares en la actualidad, como la irradiancia y la proyección de fotones, se comenzaron a utilizar también en combinación a partir de esa época.



A finales de la década de 1990 apareció también, en relación directa con los nuevos formatos de rango dinámico (HDR), introducidos por Greg Ward, a los que ya me he referido en el capítulo anterior, otros métodos alternativos que prescindían o podían prescindir de luces artificiales y utilizaban mapas de bits de alto rango dinámico como fuente de iluminación, métodos que se denominaron genéricamente IBL (*image based lighting*) y que fueron introducidos por Paul Debevec en 1997.

Durante la primera década del siglo XXI la principal innovación, desde el punto de vista teórico y práctico, ha sido la introducción de métodos de *ambient occlusion* hacia 2003.

En los últimos años, los principales avances se han producido no tanto desde un punto de vista teórico sino, principalmente, por la integración de los métodos principales anteriores en el procesamiento gestionado directamente por las tarjetas gráficas, lo que se denomina actualmente *hardware rendering*. Dado que esto último requiere una mínima incursión en la estructura de estos dispositivos, volveré brevemente a este tema en el capítulo siguiente.

Estos son los métodos principales que se utilizan en la actualidad, pues todos tienen alguna ventaja importante que compensa sus desventajas. En los apartados siguientes ampliaré esta descripción intentando reducir al mínimo los detalles técnicos para centrarme en las ideas generales que tienen bastante interés por sí mismas y, sobre todo, permitirán asimilar mucho mejor las técnicas concretas relacionadas con estos métodos.

### Ventajas y desventajas de los métodos disponibles

Desde el punto de vista de su rendimiento, los métodos disponibles se pueden clasificar en varios grupos. Una clasificación más o menos corriente es la que sigue.

a) Métodos exactos (*unbiased*) frente a métodos aproximados (*biased*). Puede decirse, simplificando bastante las cosas, que

hay métodos que convergen hacia resultados exactos y son relativamente sencillos de aplicación pues no requieren ajustar demasiados parámetros sofisticados, pero que tienen el grave inconveniente de que son muy lentos. Como decía Peter Shirley en un artículo, “con ellos se pueden alcanzar resultados exactos pero no necesariamente en vida de quien los utiliza”. Y, a la inversa, hay métodos que dan resultados más rápidos, pero a costa de introducir desviaciones con respecto a la precisión final y al uso de parámetros complejos que puede costar entender correctamente. Entre los primeros hay que citar, en primer lugar, *path tracing*. Este método da lugar a resultados precisos pero requiere un número de rayos que tiende a infinito para eliminar el ruido residual, que es la traducción visual de un número insuficiente de muestras. Hay diversas variantes de estos métodos, también denominados de “fuerza bruta”, que utilizan métodos de Montecarlo y que se basan en enviar una gran cantidad de rayos por cada píxel de la imagen que se va a obtener. Cuando un rayo se encuentra con una superficie, se vuelven a emitir rayos desde ese punto hacia lo que rodea el punto de intersección. Cuando los rayos derivados en este segundo estadio se encuentran con otras superficies se vuelven a emitir rayos, y así sucesivamente.

Entre los segundos hay que citar, en primer lugar, a *irradiance caching*, si bien en este caso casi puede decirse que la mayoría de los métodos corrientes caerían en este grupo. *Irradiance caching* se basa en tomar muestras pero no para cada píxel, como en los métodos de Montecarlo, sino para una fracción del total, tomando en consideración el hecho de que para la mayoría de las superficies, la iluminación no cambia demasiado sino que varía suavemente. Y, a partir de las muestras tomadas, se interpolan los resultados. Con *irradiance caching* se requiere un paso previo (*pre-pass*) que recolecte la información básica de la escena para tomar una selección de muestras. Esta pre-selección puede dar lugar a errores y pérdidas de detalle, y los resultados pueden incluir manchas extrañas, mientras que con fuerza bruta



todo lo que se obtiene es ruido, pequeños puntos que indican “ausencia de información”.

Las ventajas de los métodos exactos son, por tanto, que producen resultados más precisos, que sus deficiencias solo se traducen en ruido, que requieren pocos controles y que no requieren gran cantidad de memoria. Las desventajas, que son muy lentos, que no pueden computar determinados efectos (por ejemplo, cáusticas) y que no permiten grabar resultados previos que luego puedan ser refinados.

Las ventajas de los métodos aproximados son que su mayor o menor precisión puede adaptarse a los diferentes casos, con lo que pueden ser más rápidos, que en algunos casos pueden computar efectos tales como cáusticas, que permiten ajustar los requisitos de calidad y que permiten grabar resultados previos para refinarlos después. Las desventajas, que son menos precisos (aunque la precisión puede aumentarse casi indefinidamente), que sus deficiencias se traducen en varios tipos de artefactos, además de ruido, que pueden requerir ajustes específicos, que, como consecuencia de esto, requieren más controles y son más difíciles de utilizar, y que en algunos casos se puede necesitar una gran cantidad de memoria.

b) Métodos de trazado (*tracing*) frente a métodos de recolección (*gather*). Los métodos de trazado envían rayos desde las luces hacia la escena. Los de recolección envían rayos desde la cámara hacia la escena. Ambos pueden ser exactos o aproximados. Ejemplos de métodos de trazado serían *backward ray tracing*, proyección de fotones o los estadios iniciales de radiosidad. Ejemplos de métodos de recolección serían *path tracing*, irradiancia o los estadios finales de radiosidad.

Las ventajas de los métodos de trazado son que pueden simular con facilidad algunos efectos característicos propios de las luces (como cáusticas). La desventaja principal es que no tienen en cuenta la resolución ni la visibilidad de determinados objetos, por lo que se desperdicia potencia de cálculo. Las ventajas de los métodos de recolección son las contrarias: son más eficientes pues adaptan la potencia de cálculo a las características de la escena y a la resolución de salida. También permiten simular mayor diversidad de efectos que los anteriores, tales como el efecto de luces de entorno o de objetos luminosos (aunque no pueden simular cáusticas).

c) Métodos dependientes del punto de vista frente a métodos independientes del punto de vista. Ya está implicado en la distinción anterior. Los métodos de trazado son, en general, independientes del punto de vista mientras que los métodos de recolección no lo son, aunque si se utilizan en combinación con los anteriores, como es frecuente, permiten ambas alternativas.

Las ventajas y desventajas son bastante evidentes. Los métodos independientes del punto de vista no necesitan ser recalculados cuando se cambia el punto de vista. Pero, por esto mismo, son poco eficaces si el punto de vista no se va a cambiar demasiado. Y con

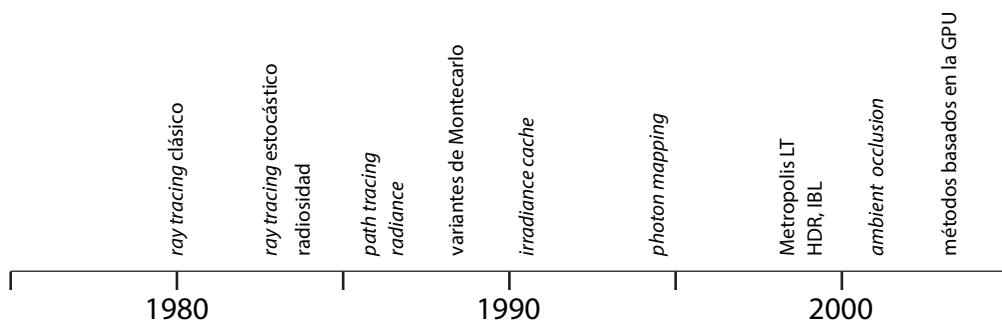


Figura 3.2 Evolución de los principales sistemas de cálculo de iluminación global.



los dependientes del punto de vista pasa lo contrario.

En la práctica, la mayoría de los métodos comerciales son híbridos y utilizan una combinación de las técnicas citadas. Por ejemplo, *mental ray* utiliza una variante de *irradiance caching* que, cuando es necesario, se combina con *photon mapping*. V-Ray utiliza varios sistemas (*irradiance caching*, *photon mapping* y variantes propias de *path tracing*), que corren a cargo de un “motor primario” y un “motor secundario” y se combinan de diferentes modos entre sí. De hecho, una de las ventajas comerciales de algunos productos (como el último citado) es que dejan al usuario que decida la mejor combinación, aunque muchos no le sacan suficiente partido porque no conocen bien el sentido de unas técnicas que resultan ser bastante sofisticadas para el usuario medio.

Por último, también hay que insistir en que, en la primera década del siglo XXI, en paralelo a la evolución de los procesadores incorporados a las tarjetas gráficas (*graphic processing units*, GPU), han surgido múltiples variantes de los métodos que se describen en este capítulo que buscan adaptarse a la arquitectura de estas tarjetas para obtener incrementos de velocidad espectaculares. Como ya he dicho, los trataré por separado en el capítulo siguiente.

### 3.3 Ray tracing clásico (1980)

#### Métodos precursores. Cálculos de visibilidad y sombras (1968, 1971)

Los precursores del *ray tracing* fueron los algoritmos para el cálculo de visibilidad y de sombras, desarrollados previamente por Apple (1968) y por Goldstein y Nagel (1971) principalmente (véase Foley, sección 15.10 para más detalles). La tesis doctoral de D. S. Kay, presentada en 1979, se basaba en métodos similares. Por otro lado, podrían citarse contribuciones a esta técnica, por esas mismas fechas, que derivaban de aplicaciones relacionadas con la simulación de trayecto-

rias de proyectiles, en balística, y a la simulación de trayectorias de partículas nucleares, en física.

Estos métodos se utilizaban inicialmente para determinar la visibilidad de los objetos en la escena, enviando un rayo para cada píxel de la imagen de salida, de tal modo que el rayo quedase determinado por dos puntos, el centro de proyección y el centro del píxel. El primer punto de intersección con un objeto de la escena establece el color del píxel correspondiente.

El pseudocódigo de este algoritmo primitivo, dado un centro de proyección C (una cámara virtual) y un formato de salida, sería algo así:

```

for cada línea de la imagen do
  for cada píxel de cada línea, P do
    (determinar el rayo desde C a P)
    for cada objeto en la escena do
      if
        (hay intersección con el objeto
         y es más cercano)
      then
        (registrar la intersección y el nombre del objeto)
        (dar a P el color del objeto registrado)
  
```

Apple (1968) utilizó este sistema para calcular las sombras de los objetos de una escena, un método que ya hemos visto en el capítulo anterior. Pero en vez de detenerse en la superficie, una vez computada la intersección del rayo con el objeto más cercano, se enviaba un rayo adicional, desde este punto, hacia cada una de las luces que hubiera en la escena. Si este segundo rayo encontraba un objeto en su camino hacia la luz, se computaba este punto como un punto que estaba en sombra. Si el objeto era semiopaco, se podía computar la atenuación correspondiente de

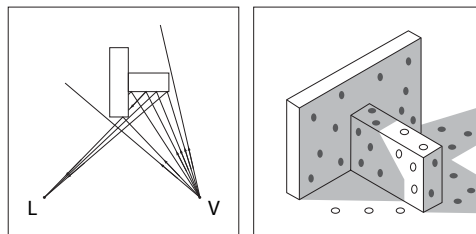


Figura 3.3 Cálculo de sombras por ray tracing.



la luz. Si no encontraba nada es que estaba iluminado.

Muchos algoritmos que computan la iluminación directa por medio de rayos de sombra aplican actualmente un esquema de integración numérica que se basa en la *rendering equation* que veremos más adelante. Pero el sistema de Apple se sigue utilizando, como ya hemos visto en el capítulo anterior, por su sencillez, precisión y rapidez de cálculo.

### Métodos clásicos de ray tracing (1980)

La mayoría de los métodos actuales de *ray tracing* son estocásticos (métodos de Monte-carlo), y cuando se habla de *ray tracing* se implica que se está hablando de estos métodos. Por otra parte, el término *ray casting*, que inicialmente era sinónimo de *ray tracing*, se reserva actualmente para la versión más primitiva de trazado de rayos, el cálculo de la visibilidad de los objetos de una escena, y en la actualidad se utiliza para una variedad de técnicas ligadas al cómputo de la visibilidad de objetos especiales o en casos especiales (por ejemplo, en videojuegos).

Por esta razón, y para evitar equívocos, los algoritmos originales, presentados por Whitted en 1980 se denominan *classic ray tracing* o *Whitted-style ray tracing*. Los métodos presentados por Whitted (1980) y por Kay (tesis doctoral de 1979) desarrollaban métodos publicados previamente por Apple (1968) o por Goldstein y Nagel (1971), principalmente, a los que me he referido en el apartado anterior.

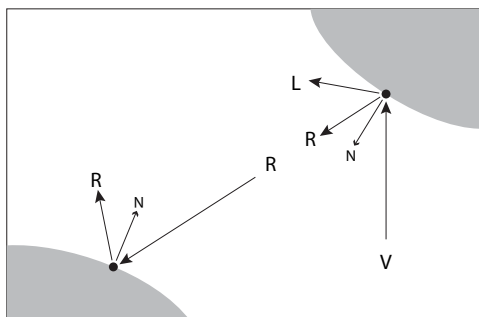


Figura 3.4 Ray tracing clásico.

Whitted y Kay extendieron dichos métodos para computar, por añadidura, reflexiones y refracciones, utilizando un procedimiento similar: por cada punto de intersección del rayo con el objeto más cercano, se enviaban dos rayos adicionales, uno correspondiente a la reflexión, computando el vector simétrico a  $V$  con respecto a la normal a la superficie  $R$  (véase la figura 3.4), en caso de que la superficie fuera reflectante, y otro  $T$  computando la desviación de  $V$  con respecto al índice de refracción de la superficie, en caso de que esta fuera transparente. Así se obtenían, para cada punto, 2 o 3 rayos secundarios,  $L$ ,  $R$  y  $T$ , que iban ramificando rayos a medida que se encontraban con nuevas superficies. Naturalmente, este método solo pudo plantearse en la medida en que la evolución del hardware admitía los centenares de miles de rayos a los que había que seguir la pista en una escena de complejidad media y que formaban lo que se denominó un *ray tree*, un árbol de rayos. El pseudocódigo de estos algoritmos primitivos es bastante más largo que el anterior (puede encontrarse en Foley, 1990, apartado 16.12).

Los algoritmos originales tenían varios inconvenientes importantes. Creaban problemas de eficiencia, pues daban lugar a miles de rayos muchos de los cuales eran innecesarios o prácticamente innecesarios para el resultado final. Creaban problemas de *aliasing* dado que las muestras se creaban a partir de una malla regular. El cálculo de las reflexiones eran preciso, pero los reflejos imprecisos que se dan en la realidad no se podían computar. Y, sobre todo, no se podía computar de modo alguno la iluminación indirecta, lo que obligaba a utilizar métodos complementarios inadecuados, como luz ambiental o luces secundarias que simulasen de un modo artificial los efectos de rebote.

### Ray tracing distribuido (1984)

Las primeras contribuciones importantes dirigidas a solucionar algunos de los problemas citados se dio a partir de la publicación de un artículo de Cook, Porter y Carpenter (1984)



en el cual se introducía la técnica que se denominó en su momento *distributed ray tracing*.

La idea principal era enviar, no uno, sino varios rayos desde cada punto de intersección, lo que permitía evitar ciertos efectos de *aliasing* y crear sombras y reflejos dispersos. La intensidad computada sería la suma de las radiancias estimadas en los puntos de impacto del trazado de rayos.

Con esto, se introducía el recurso a métodos estocásticos que se aplicarían extensamente en los años siguientes por medio de variantes diversas de métodos de Montecarlo que veremos más adelante.

En esta propuesta se incluía el cálculo de la profundidad de campo, desenfoque por movimiento (*motion blur*), sombras con umbra y penumbra y, en fin, reflejos dispersos (*glossy*). Supuso un gran avance en su momento, sin embargo, no incluía las interreflexiones difusas entre objetos como las que se lograron, por esas mismas fechas, con radiosidad.

Por otro lado, pronto resultó evidente que el método estocástico simple era muy ineficiente pues un gran número de rayos se seguían desperdiciando. A partir de la publicación del artículo de Kajiyi, en 1986, aparecieron variantes más eficientes ligadas al nuevo método de *path tracing* y variantes de

los métodos de Montecarlo, que permitirían encontrar métodos mejores de computar la iluminación indirecta, y este método quedó como un precursor importante para los que siguieron.

### 3.4 Radiosidad

#### Método general (1984)

El método de radiosidad (*radiosity*), al que también se denomina últimamente “de elementos finitos” (FE, *finite elements*), se desarrolló a partir de un algoritmo desarrollado en la Cornell University a principios de la década de 1980 y que se publicó en diferentes versiones hacia 1984. El artículo principal de referencia es el publicado por Goral, Torrance, Greenberg y Battaile (1984). Los métodos utilizados en la universidad de Cornell provenían a su vez de investigaciones llevadas a cabo a principios de la década de 1960 para simular la transferencia térmica entre superficies.

Fue el único método que se basaba en un cálculo físico correcto de la energía lumínica que se redistribuye en una escena, a partir de la energía emitida por las fuentes de luz presentes. Era, por esta razón, y, hasta cierto punto, sigue siendo, muy interesante desde el punto de vista de las aplicaciones arquitectónicas. Pero es importante conocer sus limitaciones y las alternativas disponibles, pues los inconvenientes han ido pesando cada vez y en la actualidad (2014) se utiliza cada vez menos.

A diferencia de los métodos anteriores, se basa en la geometría de la escena para calcular cómo se redistribuye la luz, en función de las propiedades de los objetos y, en consecuencia, cuál sería realmente la intensidad de cada superficie si se toman en consideración todas las interacciones entre todos los objetos de la escena. Para ello, subdivide todos los objetos en subelementos pequeños cuya contribución puede calcularse independientemente. No hay diferencia entre objetos que emiten luz y objetos que la reciben. Los pri-

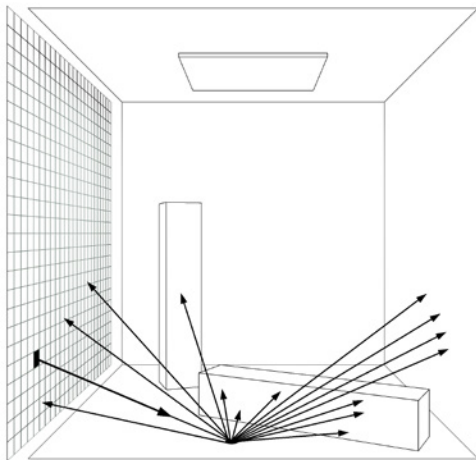


Figura 3.5 Ray tracing distribuido.



meros serán los únicos que emitirán luz al iniciar el cálculo, en una primera aproximación. Pero los segundos pasarán a ser emisores, en una segunda aproximación.

El concepto básico es la *radiosidad* que, como hemos visto en el capítulo anterior, equivale a la *exitancia*: la energía por unidad de área que surge de un subelemento o parche (*patch*) poligonal. Pero la radiosidad de un elemento se calcula a partir de la energía que emite por sí mismo más la que recibe de todos los elementos que le rodean. Es decir, la suma de la energía emitida más la energía reflejada según la fórmula siguiente:

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij}$$

en donde  $B$  es la radiosidad del elemento  $i$  (el parche considerado),  $E$ , es la energía emitida por el elemento  $i$ , y el segundo término de la parte derecha de la ecuación es la energía reflejada, que será igual a la suma de toda la energía que recibe de todos los parches que le rodean, multiplicada por el factor de forma  $F$  y por el porcentaje de reflexividad,  $\rho$ . El factor de forma depende de la distancia y de la orientación: dos objetos cercanos y enfrentados se transmitirán la máxima energía, que irá disminuyendo a medida que se alejen o giren. Para no entrar en excesivos detalles digamos que el factor de forma depende de los parámetros que se ilustran en la figura 3.7 y que intuitivamente corresponde a la distancia y la orientación de dos subelementos de la geometría de la escena.

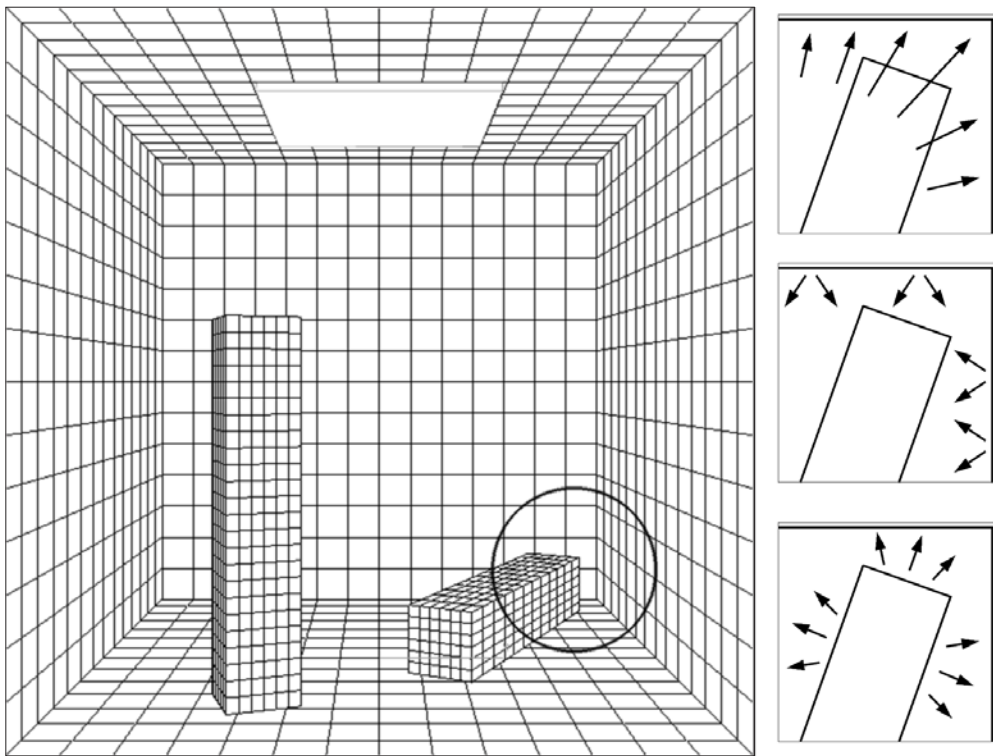


Figura 3.6 Radiosidad. Fases. En la primera fase (arriba, derecha), la única fuente de luz es la luminaria superior. En la segunda fase (centro, derecha), los parches a los que llegó la luz se convierten en emisores secundarios. En la tercera fase (abajo, derecha), los parches a los que no llegaba la luz en las fases anteriores se convierten en emisores terciarios.



Esta fórmula lleva a formar una matriz con centenares, millares o millones de ecuaciones, cuya resolución permite asignar a cada objeto una iluminación difusa determinada. Dado que la dificultad principal está en calcular el factor de forma, se fueron desarrollado métodos específicos para simplificar el cálculo de este factor.

## Técnicas de optimización

Durante los siguientes años se investigaron métodos complementarios cuya finalidad principal era reducir el tiempo de cálculo y los defectos e imprecisiones de la imagen final.

Un avance importante fue utilizar radiosity jerárquica (*hierarchical radiosity*), de tal modo que las zonas con detalles más finos se subdividían con mayor densidad, dando lugar a una malla geométrica no uniforme. Otro avance importante fue el mallado discontinuo (*discontinuity meshing*), que precomputaba mallas más precisas en zonas que se correspondían con la transición de umbra a penumbra generada por luces de área. Otro método fue la introducción de *clustering algorithms* para computar en grupos diferenciados la iluminación de objetos separados.

En 1988 se desarrolló un método denominado refinamiento progresivo (*progressive refinement*) que mostraba resultados visuales que se iban refinando progresivamente. Los métodos posteriores que se han desa-

rollado a lo largo de la década de 1990 y principios de 2000 son muy variados y, de un modo muy general, puede decirse que se dividirían en métodos de trazado (*shooting*) y métodos de recolección final (*gathering*) que, por lo general, se combinan en una misma aplicación. A finales de la década de 1990 se desarrolló una técnica denominada *stochastic relaxation radiosity* (SRR), que construía una serie de soluciones aproximadas que convergían hacia el resultado final y que utilizan actualmente muchos programas comerciales, entre ellos 3ds Max. Cito estos métodos tan solo para dar una idea de la gran diversidad de técnicas que surgieron durante aquellos años, aunque en la actualidad prácticamente se hayan abandonado. Pero, como suele ocurrir, sirvieron de inspiración para otras alternativas.

La radiosity tiene algunas limitaciones intrínsecas importantes. No puede simular directamente las reflexiones ni las refracciones, ni efectos tales como los efectos cáusticos. Y, por otro lado, necesita subdividir la geometría de la escena, lo que da lugar a archivos excesivamente pesados y a tiempos de cálculo muy elevados.

Por estas razones, los métodos de radiosity nunca se han utilizado mucho en la industria del espectáculo y en simulación visual profesional. Sin embargo, han sido bastante utilizados por los arquitectos pues tienen una ventaja muy interesante. Al estar basados en unidades fotométricas reales y al contar, por añadidura, con la posibilidad de efectuar mediciones de los resultados, es una herramienta que sirve para analizar la iluminación real de una escena y es una ayuda importante para diseñar y calcular las fuentes de luz adecuadas para un interior o, también, la iluminación que llegará a un determinado espacio en diferentes épocas y horas del año. Sin embargo, esta ventaja también ha dejado de serlo recientemente, pues los métodos basados en otras técnicas también incorporan recursos de medición equivalente, como veremos en el último capítulo.

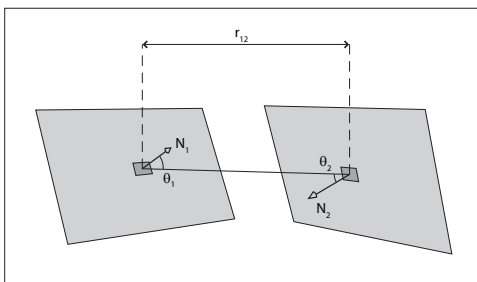


Figura 3.7 Radiosity. Parámetros básicos del factor de forma  $F$ .



## Recolección final (FG) en radiosidad

En radiosidad se suele aplicar un último paso denominado *final gather*, lo que da lugar a cierta confusión con la técnica general del mismo nombre que se utiliza en *mental ray* y que se describe más adelante. En radiosidad, este último paso consiste en llevar a cabo un cómputo, por píxel, del valor de radiancia de las superficies visibles, utilizando los valores de radiosidad almacenados en los parches superficiales generados al final del segundo pase. Hay fórmulas generales que utilizan integrales para encontrar los valores teóricos resultante, lo que no es muy eficiente en términos de cómputo. Por esto se utilizan diversas variantes de cálculo que se aproximan a este resultado de diversos modos. Dutré cita las siguientes: muestreo sobre un hemisferio uniforme; muestreo ponderado por el coseno sobre un hemisferio; muestreo sobre un área uniforme; muestreo sobre un área, ponderado por el ángulo sólido; muestreo sobre un área, ponderado por el coseno; muestreo sobre un área proporcional a la radiosidad precomputada; muestreo sobre un área proporcional a los factores de forma.

En el capítulo sobre ejemplos de configuraciones daré más detalles sobre este recurso, que mejora notablemente los resultados finales cuando se utiliza radiosidad.

## 3.5 La rendering equation

A finales de la década de 1980 había dos métodos principales de cálculo de la iluminación global. Uno de ellos, dependiente del punto de vista, era *ray tracing*. El otro, independiente del punto de vista, era radiosidad. En el primer caso, se partía de una selección de puntos infinitesimales y de direcciones dadas por la elección del punto de vista, a partir de las cuales se calculaba los valores de los píxeles. En el segundo caso, se partía de una selección de elementos discretos, finitos, y de direcciones dadas por criterios independientes de la elección del punto de vista.

Como hemos visto, ambos métodos suponían un gran avance pero eran notoriamente insuficientes, tanto desde un punto de vista teórico como práctico.

## La rendering equation (1986)

En 1986, Jim Kajiya presentó una ecuación general que describía de un modo completo el modo de calcular la intensidad de la luz emitida por una superficie en una dirección determinada, a partir de la iluminación global del entorno modelado. Aunque la referencia más conocida es la de Kajiya, en la misma fecha se presentó una ecuación similar por parte de Immel, Cohen y Greenberg (1986). Y en el primer caso, se partía de *ray tracing* para incluir la iluminación global que no se había recogido suficientemente, mientras que en el segundo se partía de cálculo global dado por radiosidad para incluir reflexión direccional que no se había recogido suficientemente. Esto era una prueba adicional de que ambos métodos, *ray tracing* y *radiosity*, eran incompletos pero que debería de haber un sistema general que los englobase.

La ecuación, en los términos en los que la presentó Kajiya, es una adaptación de principios físicos, que se remontan a las ecuaciones de electromagnetismo de Maxwell y que se han utilizado, con variaciones para diferentes aplicaciones, desde entonces. Kajiya propuso una adaptación adecuada para la computación gráfica.

La ecuación se basa en la óptica geométrica, por lo cual no permite incluir cosas tales como efectos de difracción, interferencia o polarización. Recordemos que la iluminación de una escena puede analizarse con arreglo a diferentes modelos y podemos considerar que hay tres modelos científicos que serían, en principio, pertinentes. El modelo más preciso viene dado por la óptica cuántica, que incorpora la concepción dual de la luz como onda y como partícula, a nivel subatómico, y permite hacer predicciones satisfactorias de fenómenos complejos. Pero el nivel de análisis es excesivo para aplicaciones prácticas



de simulación. Un modelo intermedio viene dado por la óptica ondulatoria, que permite incorporar el análisis de fenómenos tales como la difracción, la polarización o la dispersión, que tienen lugar cuando la luz interactúa con objetos de tamaño comparable a su longitud de onda. Este modelo se utiliza en casos especiales pero, en general, es también excesivamente complejo. El tercer modelo, el propio de la óptica geométrica, debe ser considerado como una simplificación macroscópica con respecto a los anteriores, pero permite dar explicaciones satisfactorias de fenómenos tales como la emisión, la reflexión o la refracción, asumiendo, en la mayoría de los casos, que la luz viaja en línea recta a una velocidad infinita (instantánea) y que no es influenciada por otros campos de fuerza.

La *rendering equation* es un modelo de este tercer tipo. La fórmula original, tal como aparece en el artículo de Kajiya (hay muchas variantes posteriores), es la siguiente:

$$I(x, x') = g(x, x') \left[ \varepsilon(x, x') + \int_S \rho(x, x', x'') I(x, x'') dx'' \right]$$

donde:

$I(x, x')$  representa la intensidad de la luz transportada desde el punto  $x'$  al punto  $x$ . Kajiya denomina este término "intensidad de transporte sin oclusión entre dos puntos" (*unoccluded two point transport intensity*) o intensidad transportada. En términos físicos, es la energía de radiación por unidad de tiempo y por unidad de área de la fuente y por unidad de área de su objetivo.

$g(x, x')$  es un término que representa la geometría visible. Codifica la oclusión de unos puntos por otros puntos. Si, por ejemplo, los puntos  $x, x'$  no son visibles mutuamente, este término es 0. Si son visibles, su valor es  $1/r^2$  donde  $r$  es la distancia entre ambos.

$\varepsilon(x, x')$  representa la intensidad de la luz emitida desde el punto  $x'$  al punto  $x$ . Kajiya denomina este término "emitancia de

transporte sin oclusión entre dos puntos" (*unoccluded two point transport emittance*) o "emitancia transportada". En términos físicos es la energía de radiación por unidad de tiempo y por unidad de área.

$\rho(x, x', x'')$  es un término que representa la energía dispersada por el punto  $x'$  originada por un punto  $x''$  y terminando en el punto  $x$ . Kajiya denomina este término "reflectancia de transporte sin oclusión entre tres puntos" (*unoccluded three point transport reflectance*) o reflectancia transportada.

La integral computaría así la contribución de todas las superficies de la escena, entre las que están  $x, x'$  y  $x''$ . Dicho de otro modo, la ecuación parte de unos datos de entrada (fuentes de luz, geometría de las superficies, propiedades materiales de las superficies), y obtiene unos datos de salida: los valores de radiancia en todas las superficies y todas las direcciones. Estos valores se envían al motor de *rendering* para que represente la imagen.

La ecuación formula la intensidad de la luz que llega a un punto  $x$ , como una función de la suma de la luz emitida por todos los puntos anteriores (por ejemplo,  $x_2$  hacia  $x_1$  y ha-

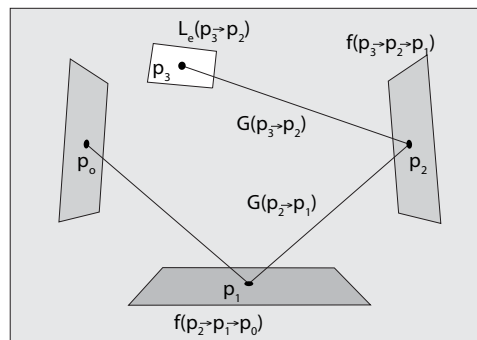


Figura 3.8 La integral sobre todos los puntos ( $p_2, p_3, \dots$ ) correspondientes a todas las superficies de la escena da la contribución total a la radiancia que deja el punto  $p_1$  en la dirección  $p_0$ . La figura muestra una superficie que emite luz ( $L_e$ ), los términos geométricos  $G$  entre los puntos que se muestran y la función  $f$  de dispersión de radiancia de las dos superficies que se muestran.





cia  $x$ ). Esto se integra hasta tomar en cuenta todas las superficies de la escena. Se considera que, para preservar la integridad del cálculo, la escena considerada está dentro de un hemisferio con las normales dirigidas hacia el interior. Por otra parte, se parte también de la base de que, como ya he dicho, la ecuación es una aproximación basada en la óptica geométrica y que, por consiguiente, descarta fenómenos propios de la óptica física tales como la difracción o la polarización, y que los materiales que intervienen tienen índices de refracción constantes.

En la literatura posterior se utiliza otra notación más genérica que es también la que seguiremos aquí:

$$L_r(p, w_r) = L_e(p, w_r) + \int_{\Omega} f(p, w_r, w_i) L_i(p, w_i) \cos \theta dw_i$$

donde  $L_r(p, w_o)$  es la radiancia de salida, reflejada, del punto  $p$  en la dirección  $w_r$ ;  $L_e$  es la radiancia emitida (si la superficie es emisora);  $L_i$  es la radiancia incidente desde la dirección  $w_i$  factorizada por la función bidireccional BRDF propia del material,  $f(p, w_o, w_i)$ , y por el coseno del ángulo formado por el rayo incidente con la normal a la superficie ( $\cos \theta$ ), integrando el resultado de esta ponderación en torno al hemisferio envolvente del punto (en el caso de una superficie transparente habría que integrar sobre la esfera envolvente).

Una resolución exacta de esta ecuación para una escena más o menos corriente requeriría un tiempo infinito, por lo que el paso siguiente, a partir de esta formulación, es discutir métodos adecuados de aproximación.

Un modo muy simple de describir la gran mayoría de sistemas actuales es afirmar que no son sino algoritmos o conjuntos de algoritmos que buscan resolver esta ecuación del modo más rápido y menos impreciso posible. Por otra parte, no hay que perder de vista que, como ya he recordado un par de veces, esta ecuación no es una descripción suficiente de lo que realmente ocurre en el mundo real pues muchos fenómenos, que desbordan el campo de la óptica geométrica,

no quedan integrados en esta fórmula, de ahí que, en determinados casos especiales, haya que complementar los algoritmos generales con algoritmos específicos para simular efectos concretos. La gran diversidad de métodos de simulación de materiales puede ser entendida como un modo de complementar este planteamiento general.

En el artículo de Kajiya ya se analizan diversos modos de resolver esta integral mediante métodos aproximados de diversos tipos que, en general, buscan minimizar el error o aumentar la precisión (reducir la variancia, en términos de teoría de las probabilidades). El método que se considera preferible es *importance sampling*. Más adelante ampliaré las referencias a este método de aproximación. Por último, se propone una alternativa a *ray tracing* que Kajiya denomina *path tracing*, en donde la idea principal es computar un solo recorrido para los rayos, habida cuenta de que uno de los principales problemas de *ray tracing* es el gran número de rayos ineficientes que es necesario registrar. Véase el apartado correspondiente a *path tracing* más adelante.

Como la ecuación no es resoluble por métodos corrientes y es necesario recurrir a métodos aproximados, nos encontramos de nuevo con una gran diversidad de métodos complementarios de optimización y variantes y hay muchos modos de clasificar estos métodos. Pero la más importante, ya mencionada, es la que distingue entre métodos que convergen hacia un resultado correcto (*unbiased*) y los que utilizan métodos de optimización que hacen que el resultado se desvíe, en mayor o menor grado, del correcto (*biased*).

La *rendering equation* es básicamente una ecuación que resuelve un problema de transferencia o transporte de energía. La energía que es emitida por una serie de luces es transportada a través de la escena por medio de reflexiones y refracciones de superficies. Al llegar a un estado de equilibrio es posible computar la energía que llega a una zona de particular interés, como es aquella en la que se forma la imagen.





## Coordenadas hemisféricas. Radiancia. Irradiancia

Los conceptos de radiancia e irradiancia aparecen como conceptos centrales en este marco teórico y en las técnicas que derivan de él, por lo que conviene revisarlo y precisar algunos términos aunque ya hemos visto, en el capítulo 1, el marco general. Doy por conocido dicho marco y solo subrayaré el papel de algunos términos principales.

En primer lugar, dado que estos conceptos están referidos al análisis de un punto de una superficie y del hemisferio que envuelve dicho punto, conviene revisar una serie de símbolos que se expresan en coordenadas hemisféricas.

En este sistema de coordenadas cada dirección viene dada por dos ángulos. El ángulo  $\Phi$  representa el azimut y se mide en relación con el eje  $x$  perpendicular a la normal de la superficie considerada. El ángulo  $\theta$  representa la elevación y se mide en relación con el eje correspondiente a la normal  $N$ .

$L_r$  es la energía emitida por un elemento de superficie  $dA$  a través del ángulo sólido  $dw$  en la dirección dada por  $\Phi, \theta$ .

$E_i$  es la energía incidente sobre un elemento de superficie  $dA$  a través del ángulo sólido  $dw$  en la dirección dada por  $\Phi, \theta$ .

La **radiancia** es el flujo por unidad de área proyectada por unidad de ángulo sólido (vatios/str.m<sup>2</sup>):

$$L = \frac{d^2\Phi}{dw dA^\perp} = \frac{d^2\Phi}{dw dA \cos\theta}$$

es decir, la radiancia representa la cantidad de flujo radiante que parte de un punto de una superficie, por unidad de ángulo sólido y por unidad de área proyectada. Es, por tanto, una cantidad de 5 dimensiones.

La radiancia tiene varias propiedades importantes que no hay que perder de vista.

En primer lugar, la radiancia es invariante a lo largo de recorridos rectos si el medio no tiene influencia apreciable. Es decir, debido

a la ley de conservación de la energía, se cumple que:

$$L(x \rightarrow y) = L(y \leftarrow x)$$

La radiancia, por consiguiente, no se atenúa con la distancia y, en consecuencia, una vez que se conoce la radiancia emitida por todos los puntos de una escena, se conoce su distribución global, lo que es tanto como decir que se puede deducir directamente la iluminación aparente en una determinada dirección.

En segundo lugar, la radiancia puede ser captada de diversos modos, por diferentes tipos de receptores, tales como los del ojo humano o los de una cámara. De lo anterior resulta que el brillo aparente de un objeto puede ser deducido de la distribución global y de la posición del receptor.

Se denomina **irradiancia** a una cantidad equivalente que permite computar la energía incidente sobre una superficie en función de la energía (potencia) por unidad de superficie (w/m<sup>2</sup>). Aunque el concepto es equivalente al anterior, se utiliza en diferentes contextos y con estrategias diferentes:

$$L = \frac{d\Phi}{dA}$$

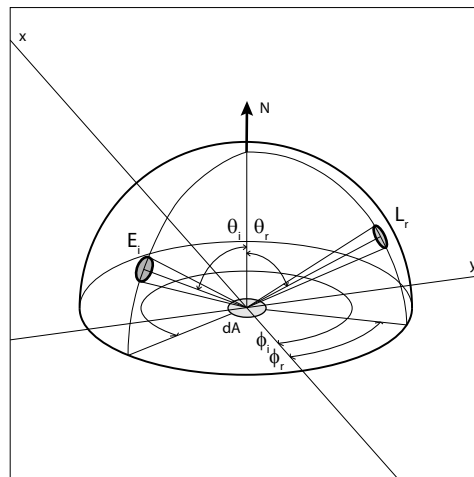


Figura 3.9 Coordenadas hemisféricas y términos básicos para las direcciones de irradiancia ( $E_i$ ), como radiación incidente y radiancia ( $L_r$ ) como radiación reflejada.

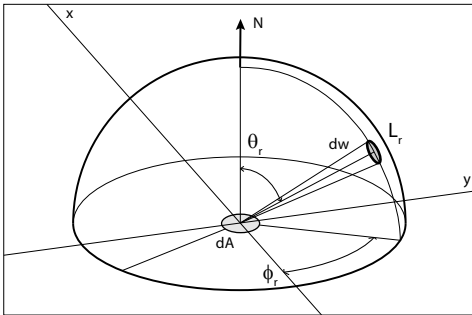


Figura 3.10 Radiancia.

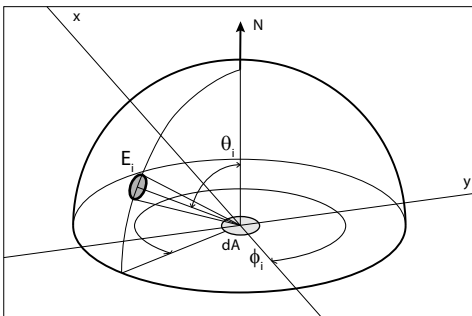


Figura 3.11 Irradiancia.

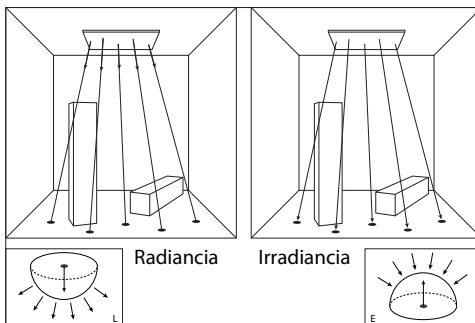


Figura 3.12 Radiancia e irradiancia.

Los dos conceptos tienden a confundirse porque una misma superficie puede evaluarse como superficie que emite energía o como superficie que recibe energía. Como se verá más adelante, el contexto permite resolver esta ambigüedad.

### La función de reflectancia (BRDF)

La función de reflectancia,  $f_r$ , se basa en la denominada “función de distribución de la reflectancia bidireccional”, o BRDF por sus siglas en inglés (*bidirectional reflection distribution function*), una función de distribución que describe la probabilidad de que un rayo de luz incidente, con una determinada orientación, se disperse en una determinada dirección de salida, con otra orientación. El término bidireccional se refiere a las dos direcciones involucradas. La figura 3.13 muestra estas dos direcciones y los dos ángulos implicados para cada una de ellas.

Esta función se utiliza principalmente en simulación de materiales, pues es la función más general que nos permite analizar cómo se comporta la superficie de un material con respecto a la luz que recibe. Pero es igualmente un elemento fundamental en el cálculo de iluminación, pues los rayos que se envían desde las luces o desde la cámara se reenvían al alcanzar un punto de una superficie según la información proporcionada por esta función. La BRDF permite computar, para cada ángulo de incidencia, la cantidad de luz que se dispersa en todas las direcciones tras ser reflejada por la superficie. Cada superficie de una escena virtual incorpora una función de este tipo que informa al sistema sobre cómo deben redistribuirse los rayos que llegan a dicha superficie. La mayoría de los sistemas actuales incorporan alguna forma de método de Montecarlo por el que se escoge una muestra significativa de la distribución ideal dada por esta función.

La función BRDF se expresa generalmente como la relación diferencial entre dos cantidades:  $L_r$  que representaría la radiancia refle-



jada, y  $E_r$ , que representaría la irradiancia, el flujo incidente por unidad de área por unidad de ángulo sólido. La BRDF es una función que implica a cuatro variables. La formulación más corriente, para una determinada longitud de onda, sería la siguiente:

$$f_r(w_i, w_r) = f_r(\theta_i, \Phi_i, \theta_r, \Phi_r) = dL_r(w_r) / dE_i(w_i)$$

donde  $w_i$  es el vector que representa la irradiancia incidente en la superficie desde una determinada dirección y  $w_r$ , el vector que representa la radiancia reflejada que sale de un punto de una superficie en una determinada dirección. O bien (segundo término) puede darse en función de los ángulos incidente y reflejado,  $\theta_i, \theta_r$ , o de los flujos incidente y reflejado,  $\Phi_i, \Phi_r$ . O bien (tercer término), en función del diferencial de la cantidad de luz reflejada,  $L(w_r)$ , y la cantidad de luz incidente  $E(w_i)$ .

La función BRDF cumple varias propiedades importantes. La primera es que si se intercambian los valores de  $w_i$  y  $w_r$ , se llega al mismo resultado. Esto es lo que se denomina la “reciprocidad de Helmholtz” al haber sido observado por Helmholtz, a mediados del siglo XIX (y es la base de una serie de técnicas actuales de reconocimiento de objetos que se denominan globalmente *Helmholtz stereopsis*). Una segunda propiedad importante de esta función es que, por la ley de conservación de la energía, la suma normalizada de las intensidades incidentes y reflejadas debe ser  $\leq 1$ . No debe perderse de vista que esta formulación general es válida para una determinada longitud de onda, que prescinde de la polarización y que considera una superficie isotrópica, es decir, que mantiene sus propiedades de reflectancia para cualquier dirección. Esto excluye muchas superficies anisotrópicas importantes para las que, como se analiza en el libro sobre simulación de materiales citado en la introducción, habrá que introducir parámetros adicionales.

Desde un punto de vista más simple, no está de más recordar que todos los modelos puede considerarse que están entre dos extremos ideales: la especularidad perfecta y la difusión perfecta. En los casos intermedios, y según los tipos de materiales, la reflectancia

varía con el ángulo de incidencia tal como ya observó Galileo y demostró y formuló Fresnel a principios del siglo XIX. En el libro de materiales se analizan con mayor detalle los diferentes modelos (Phong, Blinn, Cook y Torrance, Poulin y Fournier, Ward, Oren y Nayar, Lafortune, Ashikhmin, Shirley...) que, entre 1975 y 2000, aproximadamente se han ido proponiendo para esta función. La figura 3.14 muestra los dos extremos junto a una versión esquemática de un posible caso intermedio.

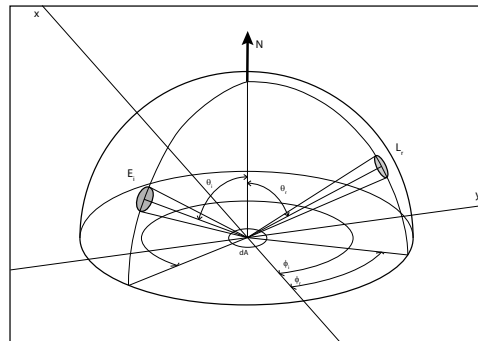


Figura 3.13 La función bidireccional de reflectancia BRDF con las dos direcciones correspondientes a la energía incidente,  $E$  y reflejada,  $L$ , y los cuatro ángulos implicados.

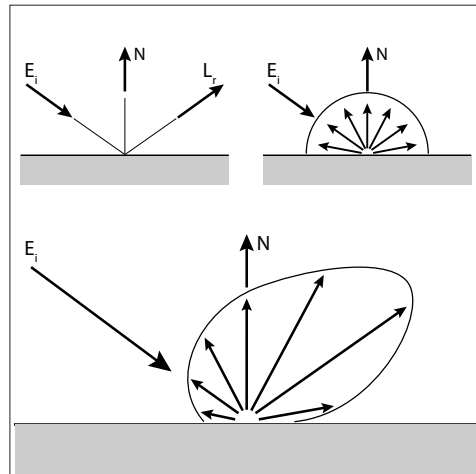


Figura 3.14 Esquema característico de la función BRDF (abajo) y dos casos extremos (arriba): especularidad perfecta y difusión perfecta.



Todos los modelos virtuales de simulación utilizan alguna variante simplificada de la función BRDF. Según el tipo de simplificación propuesta podemos clasificar estos modelos en tres categorías principales: a) Los modelos analíticos empíricos que se basan en la introducción de parámetros sin significado físico preciso pero que pueden ser ajustados para obtener resultados adecuados. La ventaja principal de este tipo de modelos es que son más sencillos de calcular y resultan más eficaces en la práctica. Su principal desventaja es que no son correctos desde el punto de vista físico; b) Los modelos analíticos físicos que se basan en la introducción de parámetros con significado físico preciso, basados en alguna versión más o menos simplificada de ecuaciones coherentes (en la práctica se eliminan algunos parámetros o se introducen algunas constantes adecuadas para hacer el modelo más manejable). Las ventajas y desventajas son exactamente las contrarias que en el caso anterior: son más precisos pero menos prácticos; c) Los basados en mediciones de materiales reales, almacenando los resultados en tablas de valores. Estos modelos se basan en la utilización de aparatos especiales que permiten tomar medidas en diferentes direcciones y almacenar los resultados en tablas. Tienen un gran interés tanto como referencia para comprobar la eficacia de los modelos anteriores como porque proporcionan una vía alternativa que puede tener una importancia creciente en el futuro.

En los sistemas de simulación de la iluminación que estamos analizando se utilizan casi exclusivamente los primeros, los modelos empíricos más sencillos.

### **Path tracing (1986)**

El método conocido como *path tracing* fue propuesto por Jim Kajiya en el mismo artículo (Kajiya, 1986) en que presentó la *rendering equation*. En lo que sigue utilizaré el término para referirme a los métodos modernos que se emparentan con los métodos clásicos de *ray tracing*, en la medida en que se basan fun-

damentalmente en enviar rayos a la escena y generar recorridos de transporte de luz (*light transport paths*) que unen luces y puntos seleccionados sobre las superficies de la escena, a partir de los cuales se pueden computar valores de radiancia. Pero los métodos modernos se diferencian de estos métodos clásicos, y también de su primera utilización por Kajiya, porque emplean toda una batería de métodos de optimización y porque tampoco utilizan exclusivamente la cámara como punto de partida, sino que pueden partir de la luz o luces, o de ambos.

En su formulación original, la principal diferencia del método propuesto por Kajiya con *distributed ray tracing* (1984) es que, en lugar de bifurcar el rayo al encontrar un punto de intersección, solo se considera un rayo de salida, sea de reflexión, sea de refracción. El argumento principal presentado por Kajiya es que uno de los problemas principales de *ray tracing* era el gran número de rayos ineficientes que es necesario registrar. Y, de hecho, la primera generación de rayos es la principal en términos de la precisión del cálculo y de los valores que aporta a la integral final. También se diferencia de *distributed ray tracing* en que el rayo que se envía está basado en elecciones fundadas en funciones de distribución adecuadas para los diferentes casos.

Esto da lugar a un recorrido lineal único (*linear path*), con un rayo que sigue un recorrido orientado por métodos probabilísticos y que va rebotando en las superficies que encuentra hasta que da con una luz, se supera el número de rebotes especificado (por un parámetro previo) o se pierde en el vacío (en caso de que no se trate de una escena cerrada). Cuando se llega este punto la función finaliza la recursión.

El objetivo es, por tanto, computar el valor de radiancia que llega a cada píxel de la imagen final. Expresado simbólicamente de un modo similar al utilizado por Dutré *et al.* (2006, sección 5.2), esto equivaldría a resolver la siguiente integral sobre el plano de la imagen:



$$L_{px} = \int_{img} L(x \rightarrow ojo) h(p) dp$$

en donde  $L_{px}$  es el valor de radiancia que llega al píxel  $p$  del plano de la imagen y  $h(p)$ , una función de ponderación que ajusta este valor sobre el plano de la imagen. Para evaluar  $p$  se envía un rayo desde el ojo que encuentra su primera superficie en  $x$ . La radiancia emitida desde el punto  $x$  en la dirección  $L(x \rightarrow ojo)$  se utiliza, en principio, para calcular el valor de  $p$ .

Este es el planteamiento original. A partir de ahí aparecen numerosas variantes que iremos viendo en los siguientes apartados

El algoritmo general se puede dividir en tres partes principales que afectan a la precisión del resultado:

- a) Número de rayos enviados a la escena por píxel. Cuantos más rayos se envíen por cada uno de los píxeles que compondrán la escena final, más exacto será el resultado pero más costosa la computación.
- b) Cálculo de la iluminación directa. En este caso, la aplicación del proceso descrito en el apartado anterior requiere decidir fundamentalmente dos cosas que afectarán igualmente a la precisión del resultado a costa del tiempo de computación: i) el número de rayos de sombra enviados desde cada punto  $p$ ; ii) el tipo de distribución escogida sobre cada fuente de luz. A esto puede añadirse, en escenas con muchas fuentes de luz, el método escogido para seleccionar la luz para cada rayo de sombra.
- c) Cálculo de la iluminación indirecta. Como en el caso anterior, será necesario tomar tres decisiones que afectarán críticamente al resultado: i) el número de rayos distribuidos sobre el hemisferio centrado sobre el punto  $p$ ; ii) el tipo de distribución escogida; iii) la probabilidad de absorción (ligada a métodos de optimización), que permitirá detener la recursión.

*Path tracing* da en la actualidad los resultados más exactos cuando se utiliza con un número suficiente de rayos. Hasta tal punto

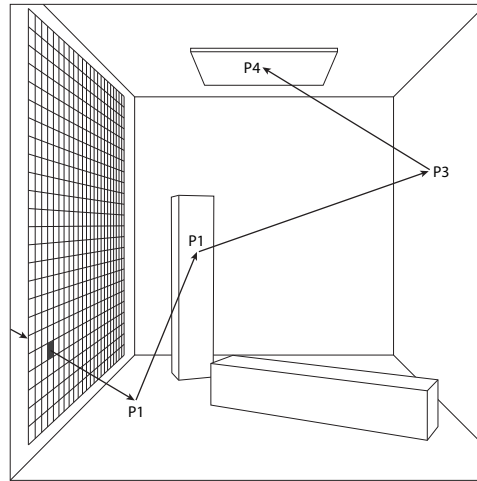


Figura 3.15 Path tracing.

es así que se utilizan como referencia imágenes obtenidas por *path tracing* para evaluar la precisión de otros métodos. La pregunta obvia, si esto es así, es ¿entonces por qué no se utiliza directamente *path tracing* y por qué no está incorporado a los programas comerciales más utilizados? Y la respuesta es: que es mucho más lento que cualquiera de los algoritmos surgidos posteriormente y cuyo objetivo principal consistía precisamente en obtener resultados suficientemente aproximados pero en tiempos muy inferiores. Obtener un buen resultado, prácticamente indistinguible de otro supuestamente más exacto, en 1 hora en lugar de 5 horas o 10 horas es obvio que resulta un factor determinante en la práctica.

### Estrategias de cálculo. *Forward tracing*. *Backward tracing*

La importancia teórica de la ecuación de *rendering* es que plantea el problema de un modo abstracto y muy general como un problema de transferencia de energía lumínica en un escenario determinado. La energía que emiten las luces se transfiere a los objetos y los objetos, a su vez, la retransmiten por medio de reflexiones y refracciones de diversos



tipos. El sistema se considera en equilibrio y el resultado del cálculo es una determinada distribución de valores de radiancia.

Pero dado que una de las propiedades fundamentales de la radiancia es que no se modifica (en el vacío) a lo largo de recorridos rectos, podemos considerar que un punto recibe radiancia o que emite radiancia y, por consiguiente, plantear la ecuación en términos de radiancia incidente o radiancia emitida.

Es posible distribuir la radiancia desde las luces hacia la escena y recoger los valores en conjuntos de los que se computan los flujos. O bien, a la inversa, los valores más importantes de las superficies se pueden distribuir y ser recogidos en las luces. También es posible tomar en consideración las dos vías y calcular su intersección. Por otro lado, la ecuación de transferencia de luz se puede escribir considerando la integración sobre un hemisferio o una esfera, situadas sobre el punto de análisis o considerando la integración sobre las superficies que rodean el punto.

Si consideramos el caso más elemental, tres puntos  $p$ ,  $p'$ ,  $p''$ , y queremos computar esta transferencia, la ecuación de *rendering* quedaría de la forma siguiente:

$$L(p' \rightarrow p) = L_e(p' \rightarrow p) + \int_A f(p'' \rightarrow p' \rightarrow p) L(p'' \rightarrow p') G(p'' \rightarrow p') dA(p'')$$

donde  $G$  es un “término geométrico” que integra una función de visibilidad (en otras formulaciones se incluye en la fórmula anterior independientemente) que computa si los puntos son mutuamente visibles y los relaciona en función de sus ángulos respectivos ( $\theta$ ,  $\theta'$ ) y de la distancia  $r$  entre ellos, con arreglo a la fórmula siguiente:

$$G(p \leftrightarrow p') = V(p \leftrightarrow p') \frac{\cos \theta \cos \theta'}{r_{pp'}^2}$$

Las dos ecuaciones son equivalentes pero representan modos diferentes de abordar el problema de la integración de la transferencia de luz. Para evaluar la formulación inicial de la *rendering equation*,  $L_r(p, w_r)$ , con el método de Montecarlo se necesita tomar muestras

aleatorias de una serie de direcciones sobre el hemisferio (o la esfera si hay transparencia) que rodea al punto, a partir de una determinada distribución sobre el hemisferio y enviar rayos en esas direcciones para evaluar la integral.

Para evaluar la formulación alternativa, se requiere tomar muestras aleatorias de una serie de puntos sobre superficies que rodean el punto, a partir de una determinada distribución sobre las superficies, y enviar rayos que conecten esos puntos, evaluando el término de visibilidad  $V$  y acumulando los puntos en un *path* de longitud creciente. La figura 3.15, por ejemplo, muestra solo cuatro puntos que están conectados por tres segmentos. Los dos primeros puntos,  $p_0$  ( $p$  en la fórmula anterior)  $p_1(p')$ , que corresponden respectivamente a la cámara y al punto dado por la intersección de un rayo enviado desde la cámara a través de un píxel y que encuentra una primera superficie, están determinados por la secuencia de barrido desde la cámara. Los otros dos,  $p_2(p'')$  y  $p_3$ , se han escogido aleatoriamente a partir de una determinada función de distribución y un algoritmo de Montecarlo. Si consideramos todos los puntos posibles de una escena tendríamos que

$$L(p_1 \rightarrow p_0) = \sum_{i=1}^{\infty} P(pp_i)$$

donde  $P(pp_i)$  computa la cantidad de radiancia que se transmite sobre un *path*  $pp_i$  con  $i + 1$  vértices. A partir de esta ecuación puede computarse la radiancia que llega al punto  $p_0$  debida a los *paths* de longitud  $i$ .

Como he dicho, este *path* puede generarse comenzando por la cámara en cuyo caso se denomina *forward ray tracing* (también denominado a veces *eye ray tracing*). En este caso, la *importance function*, una función selectiva que veremos más adelante, propaga las partes de las superficies que son visibles desde la cámara o, si se prefiere, desde cada píxel. En la mayoría de las aplicaciones del método esto no es explícito sino que está implicado en el proceso de cálculo.





El *path* también puede generarse comenzando por la luz, en cuyo caso se denomina *backwards ray tracing* (también denominado a veces *light tracing*). Este algoritmo es dual del anterior. Lo que se propaga en este caso es la radiancia enviada desde la luz que encuentra superficies visibles desde la cámara.

Escoger uno u otro método afecta al modo en que se ponderaría el método de Montecarlo pero el resultado final sería el mismo pues la radiancia no cambia. De hecho, los términos principales de la ecuación, la función de reflectancia ( $f_r$ ) y la función de forma geométrica ( $G$ ), son simétricos y se pueden intercambiar en la ecuación. Sin embargo, la metodología y los recursos de optimización varían, lo que lleva a que, como veremos, ambos métodos tengan ventajas y desventajas características. La figura 3.16 muestra un recorrido (*path*) en el que se superponen las soluciones *forward* (desde la cámara) y *backward* (desde la luz).

Además, ambas ecuaciones pueden dividirse en diferentes partes que luego se combinan para llegar a un resultado unitario. Los métodos *multipass* utilizan este recurso de diferentes modos para llegar al resultado final.

Hay una tercera posibilidad, más compleja, que es comenzar por ambos extremos y computar la intersección entre ambos recorridos, *bidirectional path tracing*. En este caso se propagan ambas cantidades al mismo tiempo. En los métodos avanzados, se computa una media ponderada de todos los posibles productos internos y todas las posibles interacciones. Esto requiere pasos dobles por lo que lo veremos más adelante, en el subcapítulo dedicado a este tema.

Sin embargo, como introducción a estos métodos más complejos, hay que decir que estas técnicas ya estaban implicadas en las técnicas que surgieron durante estos años. Enviar rayos desde la luz solucionaba un problema característico de los algoritmos basados en *ray tracing* desde la cámara dado que muchos rayos que se desviaban de su camino inicial debido a reflexiones y refracciones nunca acababan por llegar a ninguna luz, con

lo que no se obtenía información alguna y se desperdiciaba tiempo de computación. Pero enviar rayos desde la luz planteaba, a su vez, nuevos problemas debido a que un gran número de rayos no llegarían a la escena lo cual obligaba a utilizar también algún método complementario de control.

Por esta razón, en general este método se ha utilizado como complemento de *ray tracing*. Entre los primeros que aplicaron este tipo de métodos está James Arvo (1986), a partir de sugerencias avanzadas por Heckbert y Hanrahan (1984), que lo utilizaban como un preproceso junto con métodos para eliminar del cómputo con facilidad los rayos que no llegarían a interactuar con los objetos de la escena. El sistema denominado *Backward tracing* (Arvo, 1986) es, de hecho, un sistema bidireccional pues se utilizan los rayos enviados desde la luz como un pre-proceso. A cada rayo se le asigna, en este pre-proceso, una determinada cantidad de “energía” que va “depositando” en cada superficie que encuentra, en función de las propiedades de absorción y reflexión de la superficie. Y se envían tantos rayos como luces haya en la escena. Finalmente, se hace un *render* con un sistema de *ray tracing* utilizando un *shader* que computa la contribución del pre-proceso para cada punto. Así, las áreas sobre las que se “ha depositado” más “energía”, después de sucesivos rebotes, resultarán más luminosas que las otras con lo que se tiene en cuenta la iluminación indirecta.

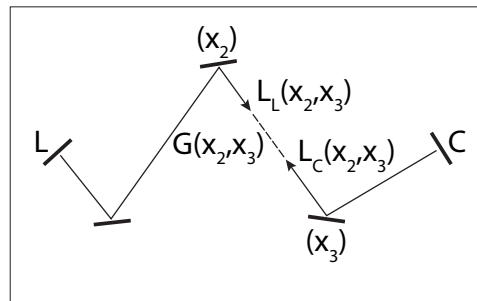


Figura 3.16 Forward tracing. Backward tracing. Los recorridos desde la cámara (C) y la luz (L) están superpuestos en la figura.





¿Cómo se comunica al *shader* la información sobre esta “energía que se ha depositado” en las superficies? O, para decirlo en los términos del artículo original de Arvo ¿cómo hacer para que los rayos que van hacia delante (*forward rays*) y los que van hacia atrás (*backward rays*) se encuentren en el medio (*meet in the middle*)? La solución que se propuso en ese artículo, y que anticipa métodos posteriores, es construir un mapa de iluminación (*illumination map*) para cada superficie. Un mapa de iluminación es como una textura pero que solo muestra diferencias de intensidad. Es un array rectangular de puntos, *uv*, a los que se asigna valores *rgb* sobre un cuadrado de  $1 \times 1$  que se proyecta mediante una función de parametrización sobre una superficie. Al comenzar el proceso se inicializan todos los valores a 0. Cuando el rayo de luz llega a  $u, v$ , se guarda en esa posición la intensidad de iluminación correspondiente. De este modo, al terminar el pre-proceso se tiene una “textura” (un *light map*, diríamos ahora) con la información requerida que luego se pasará al *ray tracing* final durante el proceso de *render*.

Esta propuesta inicial daba lugar a varios problemas técnicos, los más graves derivados de que estaba basado en muestreo por puntos, lo que daba lugar a varias formas de *aliasing*.

Por otro lado, en 1994, Philip Dutré e Yves D. Willems (Dutré, 1994), propusieron un algoritmo que utilizaba *light tracing* optimizado por medio de funciones de muestreo adaptativas que permitían construir funciones de distribución que redirigían los rayos de luz de un modo más eficiente hacia la cámara. Resumen sumariamente este método a partir del resumen suministrado por el propio Dutré en el libro publicado en 2006 (Dutré *et al.*, 2006, 5.7.1).

El algoritmo se basa en la *measurement equation* y en la *response equation* que veremos en el apartado siguiente. A partir de estas ecuaciones, el algoritmo propuesto evalúa la *measurement equation* por métodos de Montecarlo, generando rayos desde la luz en

direcciones dadas por una distribución orientada por la *importance equation*, es decir, para direcciones en las que los rayos pueden encontrar superficies significativas. La evaluación es similar a las utilizadas en métodos estocásticos de *ray tracing* optimizados. En estos métodos, se enviaba un rayo de sombra hacia la luz por cada punto encontrado para evaluar la iluminación directa. Lo equivalente en el algoritmo de Dutré y Willems es, para cada punto encontrado, enviar un rayo hacia el ojo virtual / cámara para comprobar si pasa por el píxel que se está analizando. Si es así, la contribución se registra y en caso contrario se descarta.

El algoritmo es poco eficiente pues hay que repetirlo para cada píxel. Pero puede procesarse en paralelo. La imagen se va formando de un modo gradual, bastante característico.

Como ya he dicho, lo más efectivo acabaría por ser una combinación de diferentes métodos, como veremos más adelante.

### **Función de respuesta (*response*, *importance function*). *Measurement equation***

La finalidad de la ecuación de *rendering* es obtener valores de radiancia, a partir de los cuales se obtendrán los colores de los píxeles de la imagen de salida. Cada píxel actúa como un sensor que convierte estos valores de entrada a valores adecuados de salida. Debe tenerse bien presente que, a) la respuesta del sensor es proporcional a la radiancia de la superficie visible por el sensor, b) la radiancia no varía con la distancia.

La función de respuesta (*response function*) es la encargada de llevar a cabo esta conversión. El valor de esta función,  $W(p, w)$  condensa dos características fundamentales: la geometría de la escena que rodea a un punto y las propiedades reflectivas de los materiales que conforman esa geometría.

Dado que tan solo los valores que alcanzan a los sensores son los que importan, esta función también se denomina *importance* o *importance function* por algunos autores, para enfatizar el hecho de que, para la imagen de



salida, habrá unos valores que “importan” más que otros; de entrada, todos aquellos que contribuyan a la visibilidad final y, de estos, los que tengan una radiancia superior a un umbral determinado. Esto puede inducir a cierta confusión con *importance sampling*, que veremos más adelante, que se utiliza en otro sentido técnico (como modo de ajuste de la función de probabilidad de la BRDF), aunque la idea básica sea similar.

La función de respuesta es similar en su estructura a la *rendering equation*. Pero conceptualmente es diferente pues el flujo de esta función de respuesta fluye en dirección opuesta a la de la radiancia.

Consideremos dos superficies  $m, n$ , visibles desde la cámara. Y supongamos que la superficie  $m$  refleja luz hacia la superficie  $n$ . La función de respuesta  $W(m)$  capturará la importancia de la superficie  $m$  para la imagen final y la función  $W(n)$  la de  $n$ . Pero la importancia de  $n$  lleva a la de  $m$ , que resulta ser, por tanto, más importante para la imagen final. Esto implica que el flujo de “importancia” fluye de  $n$  a  $m$  si bien el flujo de energía fluye de  $m$  a  $n$ .

La *measurement equation* combina los valores de radiancia y los valores de importancia en una única fórmula, que relaciona la solución de la *rendering equation* con el proceso de formación de la imagen final:

$$M_j = \int W(p, w) L(p, w) \cos \theta dA(p) dw$$

donde  $M_j$  es la medida de la radiancia en el píxel  $j$ ;  $W(p, w)$  es la función de respuesta que pondera la radiancia recibida por medio de una función de filtrado alrededor del píxel considerado. Este filtrado depende, como mínimo, de algoritmos de *antialiasing*. En otros casos se pondera también la apertura de la cámara y el tiempo de exposición, valores que se incorporan a determinadas aplicaciones. Y la función de respuesta se puede encontrar formulada, en otros contextos, como  $R(x, w, t)$ , en donde  $R$  es la función de respuesta,  $x$  es el filtrado del píxel correspondiente,  $w$  el valor correspondiente a la apertura y  $t$ , el tiempo (la

velocidad de obturación de la cámara). También pueden encontrarse otros factores tales como la deformación de los rayos debida a la lente o las modificaciones debidas a la sensibilidad espectral y, en general, parámetros que permiten controlar cosas como la profundidad de campo, el desenfoque de reflexiones o refracciones, el desenfoque debido al movimiento, etc.

En cualquier caso es una función que, en general depende directamente de las orientaciones de las superficies consideradas (ponderadas por  $\cos \theta$ ) y la dirección  $w$ . Y  $L(p, w)$  es la radiancia que llega al punto  $p$  desde la dirección  $w$ .

### 3.6 Métodos de Montecarlo. Métodos generales de *ray tracing*

#### Orígenes del método de Montecarlo

Todos o prácticamente todos los métodos de iluminación global avanzada actuales utilizan, de un modo u otro, métodos de Montecarlo con diversas variantes, lo que justifica una revisión previa comenzando por los orígenes de este método.

La descripción que hace Stanislaw Ulam (1909-1984), un matemático polaco que trabajaba con Von Neumann y a quien se atribuye la invención del método o cuando menos su bautismo, es quizás la mejor introducción que se puede hacer, sin entrar en detalles técnicos. Según contó, la idea se le ocurrió cuando intentaba calcular las probabilidades de que un solitario para 52 cartas saliera bien. Después de varios intentos infructuosos, llegó a la conclusión de que un método más efectivo sería hacer el solitario 100 veces y contar el número de veces que tuviera éxito. La posibilidad de aplicar este razonamiento a simulaciones que ya comenzaban a beneficiarse de la velocidad de los nuevos ordenadores es lo que dio a Ulam y otros investigadores la idea de aplicarlo a casos más importantes, lo que hicieron con considerable éxito a mediados de la década

de 1940. El nombre viene, evidentemente, del famoso casino y del juego de la ruleta.

Otro ejemplo sencillo, y más relevante para lo que sigue, es su aplicación al cálculo de un área como la de una figura plana  $S$  (ver figura 3.17) que está situada en un área cuadrada más extensa. Para ello se crean sobre el área más extensa un número  $N$  de puntos aleatorios y se computan los  $N'$  que caen dentro del área  $S$ . El área  $S$  se aproximará a la razón  $N'/N$  tanto más cuanto mayor sea el número de puntos. Para crear esta distribución aleatoria podríamos pedir a un tirador con mala puntería que disparase desde lejos sobre el cuadrado. Si el área mayor es  $A1$  y el número de impactos sobre todo el área es  $n1$  y sobre el área buscada es  $n2$  el valor del área buscada,  $A2$  será, por una simple regla de tres,  $A2 = A1 \times n1/n2$ . Si tuviéramos un caso simple, como un cuadrado de  $2 \times 2$  (área 4) en un área de  $5 \times 5$  (área 25) y efectuaráramos 100 disparos el número de disparos que caerían sobre el área interior sería de 16 ( $100 \times 4/25$ ). Pero, a la inversa, si no supiéramos el área pero sí que el número de disparos es 16 podríamos deducir que el área es 4 ( $16 \times 25/100$ ). El mismo principio se aplica a áreas irregulares.

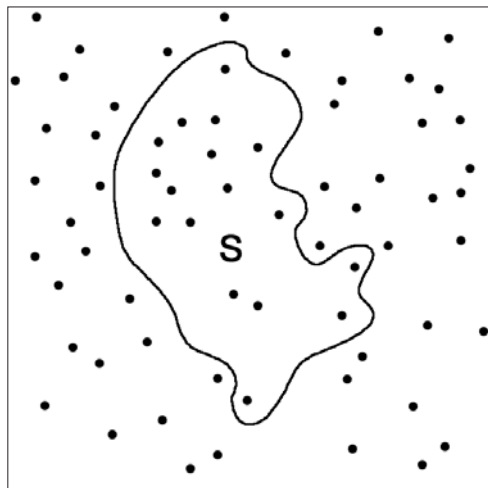


Figura 3.17 Obtención del área de una figura irregular por métodos aleatorios.

Evidentemente este resultado solo sería correcto si los puntos están distribuidos de modo uniforme y realmente aleatorio sobre la superficie. Si fueran disparos hechos por un tirador con buena puntería no serviría. De ahí que el primer problema sea conseguir una distribución uniformemente aleatoria, algo que no es tan sencillo como pudiera parecer.

Los métodos de Montecarlo, desde un punto de vista más académico, deben situarse en un doble marco: por un lado, la teoría de procesos estocásticos. Por otro lado, el desarrollo de métodos de simulación estadística de distribuciones de probabilidad a partir de la generación de secuencias de números aleatorios.

Un *proceso estocástico* es un proceso que evoluciona de modo aleatorio en el tiempo. En estos procesos intervienen una colección de variables aleatorias que toman un valor determinado según un parámetro  $t$ . El índice  $t$  se interpreta corrientemente como tiempo, y el valor  $X(t)$  designa, en consecuencia, un estado del proceso en un momento del tiempo. Si el parámetro  $t$  se puede contar, el proceso estocástico es discreto y, en caso contrario, continuo. El *espacio de estado* del proceso estocástico es el conjunto de valores que pueden tomar las variables. Por lo general, el valor de  $X(t)$  puede depender del valor anterior o de una serie de valores anteriores. Hay varios tipos de procesos estocásticos de importancia para lo que nos ocupa y se pueden clasificar inicialmente según los estados, discretos o continuos, y según los tiempos, discretos o continuos.

Una *cadena de Markov* es un tipo de proceso estocástico de particular interés, en el cual tanto los estados como los tiempos son discretos. Y en el que se cumple que la probabilidad de cada uno de ellos depende del suceso inmediato anterior pero no del conjunto de los anteriores. Fue formulada hacia 1907 por Andrei Markov (1856-1922). Si el espacio de estados viene dado por los índices  $i = 0, 1, 2 \dots$  y  $X_t = i$ , el proceso estará en el estado  $i$  en el tiempo  $t$ . La probabilidad de que esté en el estado  $X_{t+1} = j$  será  $P_{ij}$  y se denomina *pro-*



*babilidad de transición*. Los cambios posibles en un proceso de estas características se dan por medio de una matriz denominada *matriz de transición*. Se denomina *cadena homogénea de Markov* el caso en que la probabilidad de transición no depende del tiempo y “no homogénea” en caso contrario.

Las *simulaciones probabilísticas* están estrechamente ligadas al desarrollo de los ordenadores pues dependen de un gran número de pruebas y cálculos que solo han sido factibles cuando se ha empezado a contar con ordenadores potentes. La primera etapa en este tipo de simulaciones es la generación de números aleatorios que sigan ciertas distribuciones probabilísticas. Es necesario distinguir entre la generación de números aleatorios y pseudoaleatorios. La generación de números pseudoaleatorios se basa, de hecho, en algún procedimiento iterativo a partir de fórmulas lo que es, en sí mismo, determinista. La generación de números estrictamente aleatorios no es tan sencillo como pudiera parecer. Un método utilizado todavía es usar tablas de números aleatorios que están basados en métodos físicos tales como sacar números al azar de una urna o utilizar una ruleta bien equilibrada. Con la aparición de los ordenadores parecería que se pueden utilizar métodos similares, pero a menudo hay implicadas operaciones ocultas que son deterministas. El primer algoritmo para construir números pseudoaleatorios fue propuesto por Von Neumann y se conoce como el “método de centros de cuadrados”. Pero tenía insuficiencias que dieron lugar a otros métodos posteriores sobre los que hay abundante bibliografía.

La generación de números aleatorios se aplica a una determinada función. Las funciones pueden ser de múltiples tipos y dependerán de la aplicación. Es decir, a partir de estos procesos, lo que se hace es simular directamente el fenómeno físico, observar los resultados muestreados y deducir la convergencia a la que llegarían los resultados reales. El “único” requisito (en muchos casos prácticamente imposible de conseguir) es que el sistema quede adecuadamente descrito por

funciones de densidad de probabilidad o *pdf* por sus siglas en inglés (*probability density functions*). A partir de ahí todo lo que queda por hacer es obtener valores para esta función mediante entradas aleatorias y obtener los valores medios y la desviación prevista (la varianza), de tal modo que pueda estimarse adecuadamente el número de pruebas necesarias para reducir la varianza.

Aunque el método se aplica corrientemente a procesos que son estocásticos por naturaleza, también se emplea para procesos que no lo son, pero cuya resolución puede ser muy compleja y que admiten ser reformulados como procesos estocásticos. Ejemplos característicos son la resolución de algunas integrales particularmente complicadas. La idea básica es evaluar una integral compleja por medio de un muestreo aleatorio. Si se toman muestras de  $N$  puntos,  $X_1, \dots, X_N$ , según una función de densidad adecuada  $p$ , la estimación vendrá dada por una fórmula como la siguiente:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$$

El término  $F_N$  se denomina *estimator* del valor esperado de la función. El resultado, por tanto, es una variable aleatoria que depende del número de muestras  $N$  que se tomen. Para determinar la convergencia de esta integración habrá que computar la varianza de  $N$ . Puede demostrarse que la varianza decrece linealmente con  $N$  y que la convergencia es  $O(1/N^{1/2})$  en cualquier dimensión. Aunque este grado de convergencia es relativamente bajo, es suficiente para muchos problemas y, sobre todo, es susceptible de mejorarse mediante varios métodos que mencionaré más adelante, lo que supone una ventaja importante. Un rasgo importante del sistema es que el error (la varianza), es proporcional a la magnitud  $(D/N)^{1/2}$ , donde  $D$  es una constante y  $N$ , el número de pruebas. De ahí que si, por ejemplo, hay que disminuir el error 10 veces (obtener otro decimal preciso) hay que aumentar el número de pruebas  $N$  100 veces.



En determinados casos, como se demuestra en teoría de probabilidades, se pueden generar valores de un sistema por medio de cadenas de Markov. Si se puede generar una cadena de Markov que tenga una distribución estacionaria  $\pi_i$  se podrá utilizar esta cadena para estimar cantidades asociadas a  $\pi_i$ . Esta es la base de la posible aplicación del método de Montecarlo a una gran cantidad de problemas prácticos.

El método de Montecarlo tiene una estructura sencilla pues solo requiere dos operaciones principales, toma de muestras y evaluación de las muestras. Y otra ventaja importante es que es muy general, sirve para todo tipo de problemas. Para concluir, podemos decir que se basa en los siguientes componentes principales:

- La definición de función o funciones de densidad de probabilidad, *pdf* (*probability density function*) que describa adecuadamente el sistema que se quiere simular.
- Un generador de números aleatorios.
- Una regla de muestreo (*sampling rule*) que especifique el modo en que los valores generados se pasan a las *pdf* para obtener muestras parciales.
- Un método de estimación del error, esto es, una estimación de la varianza en función del número de ensayos.
- Una técnica de reducción de la varianza, esto es, algún método que permita optimizar adecuadamente el proceso o un conjunto de técnicas de optimización, de algoritmos que permitan reducir el tiempo de cálculo según el tipo de simulación requerido.

### Métodos básicos de Montecarlo en simulaciones de iluminación

En muchos problemas físicos es posible obtener modelos estadísticos de un fenómeno, a partir de los cuales se pueden intentar derivar modelos analíticos. En los ejemplos que nos importan, tenemos, por ejemplo, una luz que emite fotones que se distribuyen por todo un escenario después de rebotar en las superficies de los objetos. La luz emitirá los fotones

con mayor intensidad en unas direcciones que en otras lo que puede, en principio, asociarse a una determinada distribución de probabilidad (*pdf*). Sin embargo, si no tenemos los medios para obtener esta función y solo nos interesa llegar a una simulación razonable de la escena, se puede simular el comportamiento de la luz haciendo que emita fotones aleatoriamente, siguiendo su recorrido y recogiendo y computando sus valores. Este es el modo más sencillo de aplicar métodos de Montecarlo a la simulación de la iluminación de una escena virtual.

El procedimiento más elemental a partir de este planteamiento sería, por consiguiente, emitir fotones desde la luz y seguirles la pista hasta que ocurriera una de estas dos cosas: o bien los fotones llegan hasta la cámara y podemos computar el valor resultante (el color y la intensidad del pixel correspondiente) o bien son absorbidos y no llegan a ningún lado. Es obvio que, dado que un gran número de fotones no llegarán nunca a la cámara, deberemos enviar miles de millones para conseguir un resultado adecuado y este proceso de “fuerza bruta” elemental es muy ineficiente.

Si no se procesan suficientes muestras se genera ruido, pequeñas zonas oscuras que corresponden a la ausencia de muestras. Para eliminar el ruido solo hay dos soluciones: aumentar el número de muestras, lo que puede aumentar el tiempo de cálculo hasta límites imposibles de asumir, o bien interpolar los resultados, lo que introduce errores en el cálculo. Por tanto, el paso siguiente será buscar medios para reducir esta ineficiencia sin introducir errores importantes.

Como ya he mencionado y volveremos a ver, hay sistemas que aceptan el recurso a la interpolación y, con ello, la desviación (*bias*) con respecto a la solución correcta, y otros que aceptan imprecisiones mientras no supongan desviaciones con respecto a la solución correcta.

En la mayoría de las aplicaciones del método se generan  $N$  muestras, en direcciones aleatorias, sobre el hemisferio que rodea al punto de intersección del rayo enviado desde



la cámara con una superficie de la escena. La aleatoriedad está controlada por una función de densidad de probabilidad adecuada. La definición de esta función de probabilidad es uno de los aspectos clave que diferencian las diversas variantes técnicas del método.

La verdadera potencia de los métodos de Montecarlo reside en que permite resolver un subconjunto de la solución teórica. Y que si se escoge adecuadamente este subconjunto el resultado se acerca lo suficiente a lo que se necesita en la mayoría de los casos prácticos.

### Métodos de optimización

Hay varios métodos de reducción de la varianza o, dicho de un modo más general, de optimización del método de Montecarlo, que pueden agruparse en diferentes categorías (véase, por ejemplo, Veach, 1997, cap. 2): a) Integración analítica de una función similar a la dada. A este grupo pertenecerían métodos tales como la reducción dimensional mediante el uso de valores esperados, como *importance sampling*, o como *control variates*; b) Distribución uniforme de puntos sobre el dominio de integración. En este grupo estarían métodos tales como *stratified sampling*, *latin hypercube*, *orthogonal array sampling* o *quasi - Monte Carlo*; c) control adaptativo de la densidad de las muestras a partir de la información recogida durante el muestreo. Aquí se incluirían *adaptive sampling*, *russian roulette* y *splitting*; d) Combinación de muestras de dos o tres *estimators* cuyos valores estén correlacionados. En este grupo se incluirían métodos tales como *antithetic variates* o métodos de regresión).

Doy algún comentario adicional sobre alguno de estos métodos para que se intuya mejor el tipo de operaciones implicadas.

En estadística, se denomina *importance sampling* a una técnica para estimar las propiedades de una distribución particular a partir de otra distribución basada en un cambio de probabilidad. En simulación, es una técnica que se engloba en las técnicas analíticas

de reducción de la varianza que se usa conjuntamente con los métodos de Montecarlo. La idea principal es que determinados valores de las variables aleatorias tienen más impacto, son más significativos, que otros. En consecuencia se muestrean con más frecuencia estos valores “importantes” para reducir la varianza, lo que lleva a escoger distribuciones adecuadas que resalten estos valores. Esto puede hacer que los resultados puedan desviarse de los valores adecuados, que la estimación quede alterada (*biased*) si se aplica directamente a la simulación. Pero para evitarlo, los resultados se ponderan mediante factores que corrigen esta alteración inicial. Escoger una distribución adecuada acaba por resultar un arte pues no hay reglas que valgan para todos los casos. Es una técnica particularmente adecuada para casos en los que se presentan valores altos en una zona restringida del dominio, es decir, un dominio que presenta singularidades notorias. Veremos algunas aplicaciones de estos métodos en los próximos capítulos.

También en estadística se denomina *stratified sampling* a un método de toma de muestras que se basa en considerar subpoblaciones parciales en lugar de una población total. Esta subdivisión en estratos debe ser exclusiva (no puede haber miembros que pertenezcan a dos subpoblaciones) y exhaustiva (toda la población debe quedar cubierta por las subpoblaciones). A partir de ahí se aplican muestreos aleatorios a los estratos. Esto reduce los errores que se obtendrían con muestras de toda la población. Solo se alcanzan reducciones significativas de la varianza si los estratos tienen diferentes medias, por lo que deben ser escogidos cuidadosamente. Es adecuado para casos con un número de muestras relativamente pequeño. A diferencia de la anterior, se basa en una distribución uniforme de las muestras, al igual que otras técnicas tales como *latin hypercube sampling* (una variante de estratificación múltiple), *orthogonal array sampling* y *quasi-Monte Carlo* a la que volveré en el apartado siguiente.



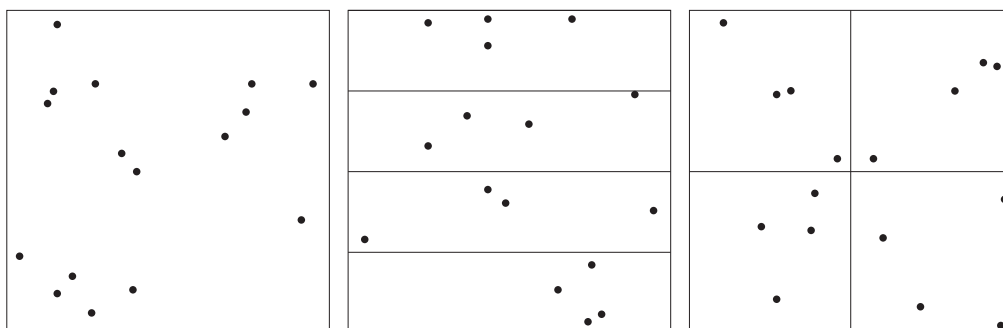


Figura 3.18 Stratified sampling: a) distribución aleatoria simple con 16 puntos, b), c) distribuciones estratificadas con 4 puntos en 4 estratos.

Otro grupo importante de métodos sería el que se puede agrupar bajo la denominación genérica de *muestreo adaptativo*. Hay dos o tres métodos principales que se pueden citar: *adaptive sampling*, *russian roulette* y *splitting*. La idea principal de *adaptive sampling* es tomar más muestras donde hay mayor variación. Esto se hace a partir de un primer muestreo que se analiza (corrientemente mediante la medición local de la variancia) para determinar las zonas que requerirían mayor número de muestras. En este sentido, es muy similar a *importance sampling*. Las diferencias son más bien técnicas y fundamentalmente se basan en que en el caso de *importance sampling* hay un análisis previo mientras que con *adaptive sampling* los ajustes se hacen dinámicamente, como parte del proceso. El principal inconveniente de este método es que los resultados introducen desviaciones (*bias*) con respecto al resultado correcto. Otro inconveniente es que funcionan bien con dimensiones lineales pero son menos efectivas con dimensiones superiores. *Russian roulette* y *splitting* son dos técnicas muy similares y que también pueden ser consideradas variantes de *importance sampling*. La idea básica de *russian roulette* es reducir el esfuerzo dedicado a evaluar muestras que no son importantes y aumentarlo cuando sí lo son. *Splitting* aumenta el número de muestras importantes para mejorar la eficiencia. A diferencia de la anterior, no introducen des-

viaciones con respecto al resultado correcto. También veremos aplicaciones de estas técnicas más adelante.

### Quasi Monte Carlo

El problema de los métodos de Montecarlo es que el espacio de búsqueda es desconocido y la búsqueda, aleatoria. Esto implica que una misma área puede explorarse varias veces y otras no ser exploradas nunca. Esto puede intuirse a simple vista en un esquema como el de la figura 3.18 (el primero por la izquierda) donde ciertas zonas presentan agrupaciones más densas y otras, más huecos.

Una alternativa es el uso de secuencias que exploren el espacio de un modo más sistemático. Estas secuencias, no aleatorias, se denominan *secuencias de baja discrepancia* (*low discrepancy sequences*). En estadística, una función de discrepancia es una función matemática que describe la cercanía de un modelo estructurado con respecto a una serie de datos dados. Valores altos indican que la correspondencia es muy pobre y viceversa. Una función de “baja discrepancia” es, por tanto, es una función tal que todos sus valores de  $N$  tienen valores de discrepancia bajos.

El uso de estas funciones da lugar a métodos que se denominan “quasi aleatorios” (*quasi random*) debido a que permiten la substitución de modelos puramente aleatorios por otros mejor distribuidos. De aquí proviene la denominación *quasi Monte Carlo* (QMC) que



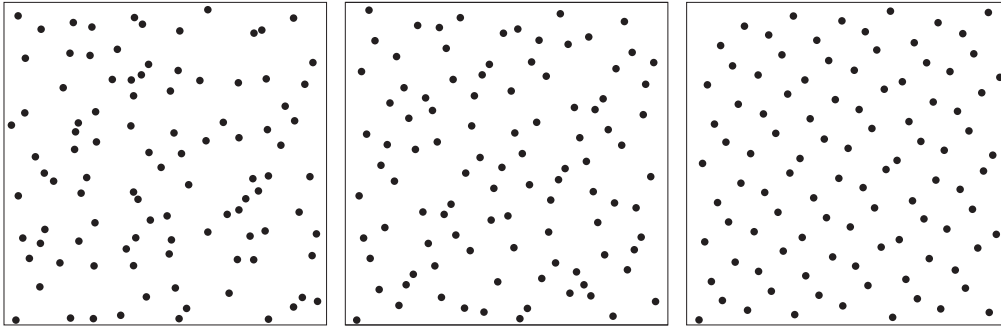


Figura 3.19 Métodos de Quasi Monte Carlo y funciones de baja discrepancia: a) (pseudo)aleatoria, b) Halton, c) Hammersley.

aparece en numerosos programas de simulación. En realidad, el término es equívoco pues no hay aleatoriedad y, por esta razón, en algunos programas se prefiere la denominación *deterministic Monte Carlo* (DMC) que no es tan equívoca pero es contradictoria. Utilizaré el primer término por la sencilla razón de que está más extendido.

Hay un gran número de secuencias de baja discrepancia que se utilizan en un gran número de aplicaciones (muchas de ellas en el mundo de las finanzas). Por orden de aparición pueden mencionarse las siguientes: la secuencia de van der Corput (1935, la más simple, equivalente a la primera dimensión de la de Halton), Halton (1960, que extiende el procedimiento a múltiples dimensiones), Hammersley (1960, similar a la de Halton), Sobol (1967), Faure (1982), Niederreiter (1987). La figura 3.19 ilustra con una serie de pocos puntos el modo en que el uso de una u otra función proporciona una distribución más regular que una distribución puramente aleatoria.

La ventaja de estos métodos es que son más eficientes, mejoran el rendimiento (el tiempo de cálculo) y pueden dar resultados más precisos. Su principal desventaja es que los errores en la estimación están directamente relacionados con la discrepancia, de ahí que los esfuerzos se hayan dirigido a crear secuencias que tengan una discrepancia baja y mantengan su efectividad con un gran número de muestras. Por otro lado, el uso de muestreos uniformes, en las simulaciones de

escenas virtuales, puede dar lugar a fenómenos de *aliasing* debido precisamente a que no son propiamente aleatorios, otro aspecto al que volveremos en el capítulo siguiente.

### Cálculo de la iluminación directa e indirecta

Un modo de reducir la ineficiencia de los métodos básicos de Montecarlo que desperdicia, en principio, un gran número de rayos que no llegan a su destino, es enviar rayos directamente hacia la luz, lo que obviamente aumentará la eficiencia.

Para que esto sea posible, la ecuación de *rendering* necesita reformularse, lo que afortunadamente ya viene dado por la propia ecuación, en la forma en que se ha presentado más arriba y que puede resumirse en

$$L_r(p, w) = L_{dir}(p, w) + L_{indir}(p, w)$$

donde  $L_{dir}$  puede representar una única luz o la suma de todas las luces de la escena. De este modo, se separan las contribuciones debidas directamente a las luces presentes en la escena de las derivadas de los rebotes en las superficies que actúan como emisores secundarios.

Si se generan muestras sobre el área cubierta por la luz y se conectan con puntos de la escena, es posible registrar los casos en que los puntos de la escena no son visibles desde la luz y asignar a estos puntos un valor de 0 en la estimación de Montecarlo, con lo



que quedan descartados del cálculo. Estos rayos se denominan “rayos de sombra” (*shadow rays*), aunque se utilicen para computar los puntos que no tienen por qué estar en sombra.

La precisión del cálculo dependerá de la estrategia a seguir que, en general, será de uno de los tres tipos siguientes: a) número fijo de rayos de sombra (un aumento en el número de rayos mejorará la precisión a costa del tiempo de cálculo y viceversa), b) número variable de rayos de sombra para cada luz en función de su importancia relativa, c) número variable de rayos de sombra para cada luz en función de su distribución propia.

Para cada luz se define una función de densidad de probabilidad que modulará el área de la luz y se utilizará por integración de Montecarlo para resolver la ecuación de *rendering*. La elección de esta función afectará a la precisión del cálculo.

Las opciones más corrientes son: a) Muestreo uniforme sobre el área de la luz. Es la opción más sencilla y que también permite utilizar mecanismos de optimización sencillos, como *stratified sampling*. Puede dar lugar a más ruido en zonas críticas, como los límites entre umbra y penumbra; b) Muestreo uniforme sobre el ángulo sólido abarcado por el área de luz. Esto da mejores resultados pues se corresponde mejor con la distribución efectiva de la luz pero es más difícil de aplicar; c) Muestreo uniforme sobre el área de luz ponderado por la intensidad de la luz. Es superior a las anteriores pero puede dar problemas en zonas débilmente iluminadas. Hay una considerable variedad de métodos alternativos. No voy a entrar en la discusión de las ventajas y desventajas de estos y otros métodos, pues basta con indicar que los problemas implicados no se resuelven de un único modo para escenas con luces extensas que requieren un cómputo preciso.

### § § §

La parte principal del cálculo corresponde al cálculo de la iluminación indirecta, esto es, a la iluminación que llega a un punto de una su-

perficie proveniente de la reflexión sobre otra superficie. Dado que hay un gran número de superficies que pueden contribuir a esta iluminación y que, a su vez, cada una de ellas puede recibir una cierta cantidad de iluminación que viene de otras superficies, el cálculo es mucho más complejo e involucra una gran cantidad de factores.

Como en el caso anterior, se parte de que la ecuación de *rendering* se puede dividir en dos partes, una correspondiente a la iluminación directa y otra, a la iluminación indirecta. La correspondiente a esta última no se puede descomponer y corresponde a 3 factores:

$$L_{indir}(p, w_0) = \int_{\Omega} f(p, w_0 w_i) L_i(p, w_i) \cos \theta dw_i$$

esto es, el producto de  $L_i(p, w_i)$ , la radiancia incidente desde otras superficies, sobre el punto  $p$  proveniente de la dirección  $w_i$ ; la factorización de este valor que viene dada por la función bidireccional  $f(p, w_0 w_i)$ , propia del material asignado a la superficie y, en tercer lugar, el coseno de  $\theta$ , el ángulo formado por el rayo incidente con la normal a la superficie. El problema consistirá, por tanto, en evaluar esta integral de un modo adecuado.

El método más general consiste en evaluar esta integral por medio de un método de Montecarlo basado en una función de distribución uniforme sobre un hemisferio centrado sobre el punto  $p$ , que toma una serie  $N$  de muestras aleatorias. Pero la evaluación de  $L_i$  requiere repetir el mismo cálculo para la superficie vecina y este, a su vez, el de la anterior con lo que el algoritmo es recursivo y el coste puede ser excesivo.

La evaluación puede detenerse, como en la descripción de los métodos estocásticos simples, mediante un método de optimización de *russian roulette*, ligada a una probabilidad de absorción apropiada para la escena que sea y que, por lo general, se liga a la reflectancia propia de la superficie considerada; o bien mediante alguno de los métodos que ya he citado. Así, en lugar de una distribución uniforme puede utilizarse *importance sampling* para construir una función de distri-



bución más adecuada basada en alguno de los tres factores que constituyen la fórmula: la función de reflectancia bidireccional (esto permitiría descartar direcciones con valores bajos de reflectancia), o el coseno del ángulo entre el rayo incidente y la superficie (esto permitiría descartar direcciones cercanas al horizonte) y la radiancia incidente (esto sería más complejo, pues requeriría una estimación previa que permitiera descartar valores incidentes bajos). Otra alternativa es tomar muestras directas del área en lugar de tomar muestras sobre el hemisferio. Esto puede aumentar la varianza pero tiene algunas ventajas desde el punto de vista del cálculo. En cualquier caso, la elección de alguno de estos métodos o de otra estrategia constituye el núcleo principal de las diferentes estrategias de cálculo que han ido evolucionando hasta llegar a combinar diferentes métodos y, como veremos en la sección siguiente, a diferentes procesos cuyos resultados se combinan en la imagen final.

### **Ray tracing generalizado. Métodos de optimización. Estructuras intermediarias**

Aunque los diferentes apartados de este capítulo están dedicados a técnicas concretas de iluminación avanzada, la mayoría de estas técnicas utilizan algún modo de trazado de rayos, *ray tracing*. De ahí que, por un lado, las técnicas basadas en *ray tracing* se han convertido en una alternativa muy importante a lo que se denominan genéricamente técnicas de rasterization y que, por otro lado, esta denominación resulte equívoca pues se confunde con uno de los métodos concretos de calcular la iluminación avanzada.

Volveremos sobre el primer tema en el capítulo siguiente, en el contexto de la estrecha relación que hay entre las técnicas de simulación y la evolución de las tarjetas gráficas, de las GPU (*graphics processing units*). Por lo que respecta al segundo tema, convendrá, en primer lugar, clarificar los términos y, en segundo lugar, dar alguna indicación aunque sea muy breve sobre una cuestión muy técnica pero de gran importancia: los diversos modos en que se optimizan los métodos generales o particulares basados en *ray tracing*.

Por lo que respecta a los términos, y como ya hemos visto, se denomina convencionalmente *ray casting* al método más elemental de enviar un rayo en dirección de un objeto para hacer un cómputo elemental, sea para encontrar un punto de intersección sea para, como en el cálculo de sombras, para enviar desde el punto de intersección otro rayo hacia la luz. Y se denomina *recursive ray tracing*, *classic ray tracing* o *Whitted style ray tracing* al procedimiento que hace derivar nuevos rayos desde el punto de intersección en función de las propiedades de reflexión o refracción. Y se denominó *distribution ray tracing* o *distributed ray tracing* (este era el término original utilizado por Cook, pero se cambió para evitar equívocos para métodos basados en el uso de ordenadores en red) a los métodos que generaban múltiples rayos desde el punto de intersección.

La diferencia principal, que surgió en los años siguientes, es que los nuevos métodos pueden leer la información incorporada en *shaders*, el nuevo concepto revolucionario que, como veremos más adelante, permitió independizar el comportamiento de las superficies (y cámaras, luces y volúmenes) del procedimiento de cálculo de la iluminación, de tal modo que la derivación del rayo viniera dada por el *shader* particular que encontrase en su camino.

En paralelo se generalizaron los métodos de Montecarlo, a su vez con múltiples variantes (*importance sampling*, *QMC*, *quasi Monte Carlo*, etc.) para aplicarlos a múltiples aplicaciones de *ray tracing* o a lo que podríamos denominar *ray tracing en sentido general*, sin adjetivos, ligándolos a funciones de distribución de diversos tipos para diferentes aplicaciones, globales o parciales.

Por lo que respecta a los métodos de optimización, la principal desventaja de *ray tracing* (el resto son ventajas) es el enorme número de intersecciones que debe calcular. Por



esta razón, los métodos de cálculo basados principalmente en *ray tracing* se han concentrado en buscar métodos de optimización del cálculo. Puede decirse que los diversos métodos se agruparían en dos vías principales: la reducción de rayos en el origen y la reducción de rayos en las derivaciones. En la descripción que sigue me baso principalmente en la dada por Ingo Wald en su tesis doctoral de 2004, en las secciones 3.1 y 3.2.

La primera vía, la reducción de rayos en el origen, implica enviar no uno o varios rayos por cada píxel sino un número reducido de muestras en el plano de la imagen. Para ello se han utilizado las alternativas siguientes:

- a) Selección de píxeles y muestreo adaptativo. Consiste en utilizar un número reducido de píxeles basándose en criterios tales como el contraste o los valores de profundidad de píxeles contiguos.
- b) Trazado de vértices. En este caso, la selección se basa en vértices visibles de la escena a los que se envían primordialmente los rayos, utilizándose después métodos de interpolación para encontrar los valores intermedios.
- c) *Render cache*. En este caso se guardan muestras de marcos anteriores y se reproyectan sobre los nuevos para reconstruir la imagen a partir de ciertos recursos heurísticos;
- d) *EPI (edge and point image)*. Es una variante de la anterior que localiza discontinuidades significativas tales como siluetas de objetos o bordes de sombras y los utiliza para mejorar la reconstrucción de la imagen.

La segunda vía, la reducción de rayos secundarios, es aún más importante pues, en general, se trazan varios rayos secundarios por cada rayo primario. Para reducir este impacto se han utilizado los recursos siguientes:

- a) *Shadow caching*. Se basa en que la mayoría de los rayos disparados por un *ray tracer* son rayos de sombra que solo requieren encontrar, o no, un objeto en su camino para comprobar que hay oclusión.

Y muchos de ellos son similares pues se encuentran con el mismo objeto. Para reducir el coste de cálculo se guarda, en un registro de acceso rápido (*cached*), en cada luz, el último objeto de oclusión y los nuevos rayos se comparan con este objeto, lo que reduce el tiempo de cómputo. El método tiene algunos inconvenientes técnicos pues la coherencia implícita no es total. Pero es probablemente el método más utilizado.

- b) *Local illumination environment*. Este método subdivide la escena en *voxels* que guardan información sobre la influencia de las luces en su región, concretamente, si la luz es visible o no visible o parcialmente visible desde el *voxel*. Solo estas últimas requieren un análisis más preciso. El principal inconveniente de este método es que requiere la construcción de esta estructura de *voxels* lo que también consume recursos si bien hay métodos adicionales que reducen este coste.
- c) *Adaptive shadow testing*. En este caso, las luces se ordenan en función de su influencia sobre el punto considerado y solo se trazan rayos hacia las más importantes;
- d) *Single sample soft shadows*. Para computar las sombras suaves debidas a luces de área se reduce la luz a un punto que luego se atenúa en función de su alcance a posibles ocluiders cercanos.
- e) *Pruning*. En escenas con muchas superficies reflectantes puede haber una “explosión” del árbol de rayos, pues si cada superficie envía varios rayos el crecimiento puede llegar a ser exponencial. Pero puede ponderarse la importancia de los rayos en función de factores de atenuación de la reflexión o refracción y también en función de su mayor o menor contribución al píxel en que se originaron. Esto permite “podar” (*pruning*) los rayos reduciendo su número. El método puede dar lugar a ruido si no se pondera bien este recorte.
- f) *Reflection mapping and shadow maps*. El uso de estos recursos puede utilizarse en *ray tracing* para aproximar efectos en lugar



de trazar rayos, aunque sea un método contradictorio con el propio procedimiento de *ray tracing*. En consecuencia, puede dar lugar a errores y artefactos diversos.

Por último, por lo que respecta a las estructuras intermediarias, hay que añadir lo siguiente.

Otro método obvio de mejorar la eficiencia de *ray tracing* es aumentar la velocidad de cálculo. Además de intentar hacer esto por medio de mejoras en el hardware también es posible por medio de mejoras en el software, puliendo los algoritmos para que sean más eficientes. Esto ha llevado principalmente a mejoras en las pruebas de intersección, por un lado, pero, sobre todo, a la investigación sobre estructuras espaciales de subdivisión jerárquica que permitan reducir el número de cálculos mediante una estimación previa. Estas “estructuras de aceleración” (*acceleration structures*) pueden ser atravesadas con mayor rapidez si su estructura se adapta adecuadamente a la escena y el algoritmo utilizado. Hay un gran número de estructuras que se han propuesto y utilizado en los últimos años. Las principales son las siguientes:

a) *BVH (bounding volume hierarchies)*.

Cada primitiva se almacena una vez en la jerarquía. La ventaja es que la memoria necesaria puede controlarse directamente y se asegura que cada primitiva será intersectada una sola vez. La desventaja es que diferentes partes de la jerarquía pueden solaparse en diferentes regiones espaciales, lo que resulta ineficiente.

b) *Subdivisión espacial por grids, octrees, BSP*. En este caso, todo el espacio se subdivide de un modo sistemático el volúmenes yuxtapuestos de diversos tipos: redes uniformes (*grids*), árboles octales (*octrees*), particiones binarias (*BSP, binary space partition*). Al atravesar la estructura, tan pronto como se encuentra una intersección a una determinada distancia todos los *voxels* situados más allá de esa distancia pueden ser descartados, lo que resulta más eficiente. La desventaja es que requiere más memoria, pues puede ser necesario llegar a subdivisiones muy densas lo que consume mucha memoria y puede ha-

cer que la misma primitiva se encuentre en varios *voxels*.

El modo en que se atraviesa estas estructuras también tiene importancia pues, dada una estructura determinada, esta puede ser recorrida de diferentes modos por diferentes algoritmos. Y la propia construcción de la estructura se lleva a cabo por otros algoritmos que también pueden ser más o menos eficientes. No podemos entrar en los detalles de estos procedimientos, pero baste con añadir que hay un gran número de trabajos dedicados a estos temas. Y la eficiencia se mide por una terminología que puede encontrarse en numerosos artículos y que clasifica el orden de complejidad computacional,  $O$ , por medio de una fórmula basada en el tamaño  $N$  de los elementos a procesar (número de datos, elementos de una matriz, nodos, etc.). Los más corrientes, de mayor a menor tiempo implicado, son  $O(\log N)$ , u orden de complejidad logarítmico,  $O(N)$  u orden lineal y  $O(N \log N)$  u orden cuasi lineal. Otros órdenes progresivamente menos eficientes son el cuadrático,  $O(n^2)$ , el cúbico,  $O(n^3)$ , etc.

### 3.7 Métodos de pases múltiples (*multipass*). Métodos *biased*

Ante las insuficiencias complementarias de los dos sistemas principales, *ray tracing* y radiosity, se fue planteando la posibilidad de integrar de algún modo los dos sistemas. Por otro lado, fue adquiriendo una importancia progresiva la combinación de diferentes métodos de Montecarlo, con diferentes estrategias y técnicas.

Durante la década de 1990 se probaron diversos métodos de mejorar los resultados por medio de pases múltiples, combinando diferentes técnicas que pasaban sus resultados a otras técnicas. Aunque estos métodos parecería, de entrada, que duplican el tiempo de cálculo necesario, se comprobó que las mejoras en eficiencia que de este modo se conseguían, compensaban este incremento.



Por otro lado, adquirieron cada vez mayor importancia métodos que se desviaban del resultado correcto (métodos *biased*) por medio de una reducción en el número de muestras y el recurso generalizado a métodos de interpolación de resultados. Pero se comprobó también que la desviación teórica de los resultados correctos era imperceptible en la mayoría de los casos y la exigencia teórica quedaba cada vez más desfasada en relación con las exigencias prácticas. El método principal, en este sentido, y el que seguramente sigue siendo más utilizado en la actualidad, es *irradiance caching*, en sus diversas versiones.

### Métodos combinados de *ray tracing* y radiosidad (1986)

Quizás el primer método combinado que se presentó de modo sistemático fue el propuesto por Wallace, Cohen y Greenberg (1987). En este artículo se utilizaba un método extendido de radiosidad que era capaz de computar los reflejos. Y este método se utilizaba como un precálculo que luego se combinaba con el cálculo final. Esto requería toda una serie de nuevas técnicas que, entre otras cosas, deberían asegurar que un mismo recorrido no se computase dos veces, lo que falsearía el resultado final.

A medida que las dificultades técnicas de estos métodos de precálculo se iban superando, su uso se fue extendiendo y, entre 1990 y 1995, los métodos más eficaces utilizaban una combinación de *ray tracing* extendido con técnicas de Montecarlo y radiosidad. Sin embargo, ambos métodos tenían inconvenientes importantes. *Ray tracing* extendido era muy lento y los resultados incorporaban ruido difícil de eliminar. Radiosidad utilizaba una gran cantidad de memoria, no podía computar las reflexiones especulares y daba resultados que incorporaban varias formas de *aliasing*.

Los métodos combinados partían de radiosidad junto con formas simplificadas de *ray tracing* para simular reflexiones especulares. En otros casos se utilizaba radiosidad combinada con *path tracing* para computar las re-

flexiones difusas directamente visibles, lo que reducía el *aliasing*. Había un buen número de métodos que utilizaban este tipo de técnicas combinadas. Puede encontrarse un resumen de alguna de estas técnicas en la introducción de Jensen (1996) a la presentación de la nueva técnica de proyección de fotones. Y en varios artículos que tratan de *multipass methods*.

En los años siguientes las ventajas de combinar diversos métodos se fueron haciendo más patentes hasta tal punto que puede decirse que la gran mayoría de los métodos surgidos a partir de 1990, aproximadamente, son métodos *multipass*.

### *Bidirectional path tracing* (1993)

El método de *bidirectional path tracing* fue propuesto por Eric P. Lafortune e Yves D. Willems en 1993 y, en paralelo, por Eric Veach y Leonidas Guibas por esas mismas fechas (Veach y Guibas, 1994, 1995), como una alternativa más eficiente a *path tracing*, dirigida principalmente a mejorar los resultados en escenas interiores en donde la iluminación indirecta es importante, por medio de una variante específica de métodos de Montecarlo.

La idea básica es que es posible combinar las dos vías características: enviar rayos desde el ojo o desde la luz, esto es, utilizar *forward path tracing (gathering)* o bien *backward path tracing (shooting)*. En este método, se generan recorridos independientes y los extremos se conectan entre sí. Por añadidura, en cada nodo o punto de intersección con las superficies, se envían rayos de sombra (*shadow rays*) que conectan los nodos respectivos de los caminos homólogos y sus contribuciones respectivas, si existen, se añaden al flujo general. De este modo, se computan las contribuciones de la iluminación indirecta.

Los paseos aleatorios así generados se computan siguiendo procedimientos de cómputo propios de los métodos de Montecarlo ya vistos. Es decir, que para cada nodo





se computa una *pdf* (*probability distribution function*) que está basada en la función de reflectancia (BRDF) propia de la superficie correspondiente, y una función de *importance sampling* que asegura que se envían más rayos en direcciones con reflectancias dominantes.

El método utiliza una estimación primaria y una estimación secundaria que reduce el ruido a partir de una ponderación de los resultados obtenidos en la primaria.

Las contribuciones de cada vía deben ser ponderadas adecuadamente. Se considera la especularidad del punto superficial correspondiente de tal modo que si la ponderación de la vías hacia el ojo correspondiente al punto  $p$  es  $W_p$ , la correspondiente a la otra vía, la de la luz, es  $1 - W_p$ . Si la superficie es muy especular, el primer valor se aproximará a 1 y el otro, a 0. Para superficies más bien mates ocurrirá lo contrario. Hay diversas complicaciones técnicas en las que no voy a entrar y que se pueden encontrar en los artículos originales citados (véanse las referencias). Pero la idea básica es sencilla y constituye un precedente importante para los métodos que seguirían. La proyección de fotones que se describirá más adelante supuso una alternativa más eficiente al *ray tracing* bidireccional pues, en lugar de repetir el cálculo desde otra dirección, la información lumínica se almacenaba ya en la primera vuelta. Y uno de los autores citados, Eric Veach, haría otra contribución fundamental y más importante pocos años más tarde, *metropolis light transport*, que veremos más adelante.

### **Irradiance cache (1988, 1992)**

Este método nació como una de las alternativas propuestas a finales de la década de 1980 para utilizar métodos más eficientes de cálculo. Los métodos de Montecarlo exigían enviar millones de rayos a la escena y muchos investigadores se plantearon si no sería posible encontrar resultados aproximados pero satisfactorios enviando menos rayos e interpolando los resultados. Esta alternativa temprana

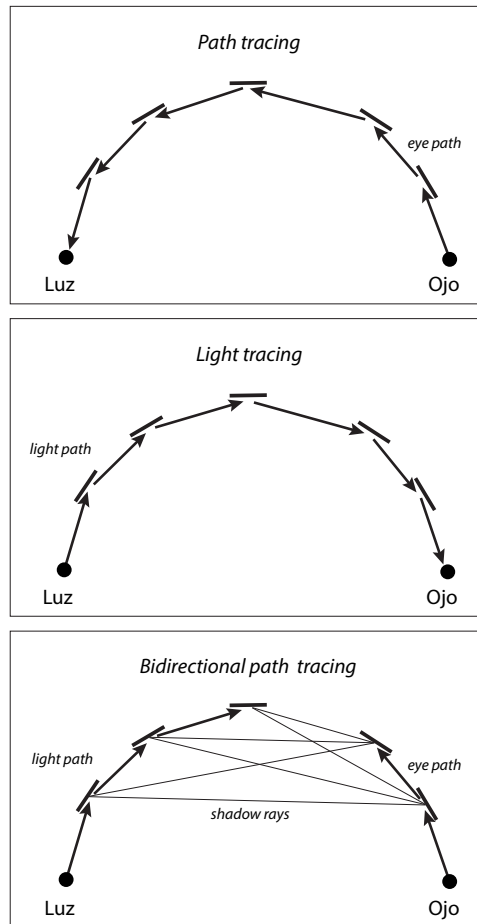


Figura 3.20 Bidireccional path tracing. Representación esquemática de los paths de la luz y el ojo y de los rayos de sombra que conectan los nodos y comparación con path tracing y light tracing.

está estrechamente relacionada con el desarrollo de un programa específico, *Radiance*, que fue desarrollado por Greg Ward durante los primeros años de la década de 1980, para ofrecer a arquitectos y diseñadores una herramienta que les permitiera hacer simulaciones suficientemente realistas de la iluminación en interiores. *Radiance* es, de hecho, un conjunto de módulos que incluyen no solo herramientas de simulación y *rendering* sino también herramientas de análisis. Entre sus numerosas aportaciones también hay que citar que,



como ya hemos visto, avanzaba las nociones fundamentales de HDR (*high dynamic range*) que se haría público y se empezaría a utilizar extensamente en la década siguiente. Se desarrolló en California (en el Lawrence Berkeley Laboratory) y en Lausane (en la École Polytechnique Fédéral), en Suiza. El software se divulgó como software libre y sigue estando a disposición de cualquiera que lo quiera descargar de la página web de *Radiance* (véanse las referencias) aunque conviene advertir que su uso e instalación no son tan sencillos como los de un programa corriente.

Estos métodos, que suponían un avance teórico y técnico con respecto a los sistemas de iluminación global que se estaban discutiendo por esas fechas, fueron publicados en un par de comunicaciones de 1988. En la principal de ellas (Ward, 1988) se remarcaba el hecho de que la luz indirecta da lugar a gradientes suaves. Y se proponía, en consecuencia, reducir el número de evaluaciones de la escena por métodos de Montecarlo y, en lugar de derivar valores directos, interpolar entre valores previamente calculados. Ward denominó a este método “evaluación perezosa” (*lazy evaluation*).

Posteriormente, a partir principalmente de 1992 (véase Ward y Heckbert, 1992), pasó

a denominarse *irradiance caching*. Este último término deriva del uso de una entrada de *cache* (una zona de la CPU, pequeña y de acceso rápido, donde se almacenan datos usados repetidamente), que almacena corrientemente la posición y la normal del punto evaluado, y que puede almacenar también los valores de irradiancia calculados para el punto, de tal modo que si la *cache* contiene valores adecuados se interpola entre ellos y, si no, se recalcula el valor y se añade a la *cache*. Estos valores así guardados (*cached*) se guardan en un mapa 3D, una nube de puntos que pasó a denominarse *irradiance map* y que guarda, por tanto, dos tipos de valores, la posición geométrica del punto muestreado y la iluminación que recibe de los puntos vecinos.

Este mapa sirve de base para el pase siguiente. En este segundo pase se deduce por interpolación la iluminación de los puntos intermedios que no están representados en el mapa, y que puede hacerse de varios modos.

Lo fundamental del método propuesto por Ward no está en el desarrollo de un técnica especialmente novedosa sino, para decirlo en sus propios términos, “en lugar de aumentar el número de muestras, como se ha hecho con algoritmos que se basan en la fuerza

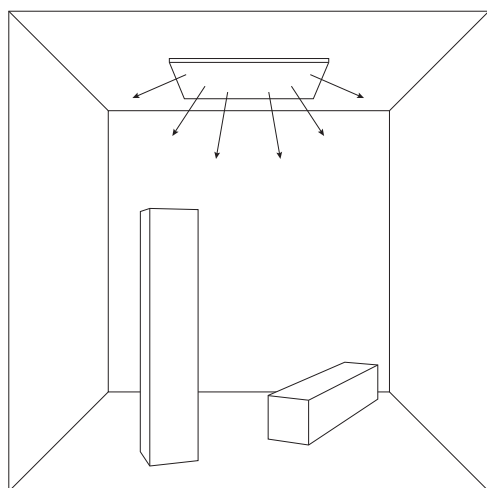
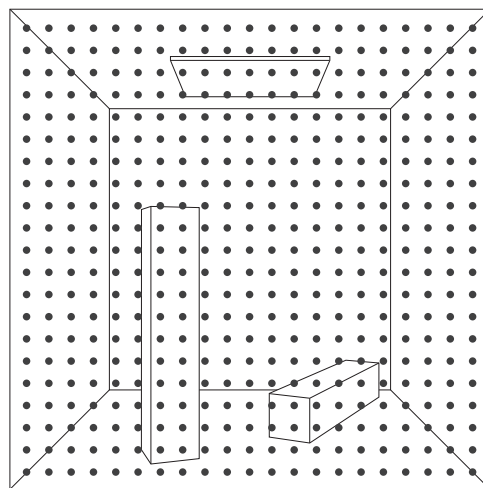


Figura 3.21 Irradiance cache. Primer paso.





bruta, nuestro método aumenta el contenido de información de cada muestra, para incluir estimaciones de la primera derivativa, o gradiente, de la irradiancia”.

*Radiance* es básicamente un método de *ray trace*, basado en algoritmos como los utilizados por Whitted, pero que utiliza *irradiance caching* para computar de un modo más eficiente la interreflexión difusa entre superficies. Utiliza *ray trace* para llegar a una evaluación recursiva de una ecuación que es básicamente la misma que la utilizada por Kajiya con la única diferencia conceptual de que, en lugar de tomar en consideración la transferencia de energía entre dos puntos, se considera la energía que pasa a través de un punto en una dirección específica. La fórmula que condensa estos principios es:

$$L_r(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^\pi L_i(\theta_i, \phi_i) f(\theta_i, \phi_i; \theta_r, \phi_r) \cos \theta_i \sin \theta_i d\theta_i d\phi_i$$

donde:

$\theta$  es el ángulo polar medido a partir de la normal a la superficie,

$\phi$  es el ángulo azimutal medido a partir de la normal a la superficie,

$L_r(\theta_r, \phi_r)$  es la radiancia reflejada,

$L_i(\theta_i, \phi_i)$  es la radiancia incidente,

$f(\theta_i, \phi_i; \theta_r, \phi_r)$  es la función BRDF.

Al igual que en la mayoría de métodos de *ray tracing*, se computan de modo independiente las contribuciones directas y especulares de cada luz. Y las contribuciones difusas indirectas se computan por medio de métodos de Montecarlo sobre el hemisferio que rodea al punto muestreado. Pero un aspecto clave de esta nueva alternativa es que, dado que esta irradiancia indirecta es independiente del punto de vista y cambia suavemente sobre la mayoría de las superficies de la mayoría de las escenas, resulta más eficiente calcularla solo cuando sea necesario, captando (*caching*) los valores computados para llevar a cabo una interpolación local.

Estos valores seleccionados no se distribuyen según una malla regular sino que se seleccionan de zonas que puedan mostrar variaciones mayores, concretamente en zonas correspondientes a superficies cercanas o con mayores curvaturas. Y también se acepta extrapolar valores si esta extrapolación no tiene lugar en zonas críticas. De este modo se generan *gradientes de irradiancia* (*irradiance gradients*).

El meollo del método está en conseguir valores adecuados de muestras seleccionadas adecuadamente. Estas muestras, que consisten en un punto, un vector normal y un valor de irradiancia, se almacenan en un árbol octal para ser interpoladas posteriormente. El cómputo se basa en dos vectores, uno, el gradiente rotacional (*rotational gradient*) y otro, el gradiente traslacional (*translational gradient*) que vienen dados por su dirección y magnitud. Ambos están situados en el plano base del hemisferio (tangente a la muestra) que discurre a lo largo de una superficie determinada. Así se obtiene una información fundamental acerca de la distribución básica de la iluminación sobre una superficie determinada.

El método propuesto por Ward se ha ido revelando a lo largo de los años, solo o en combinación con otros métodos, de una

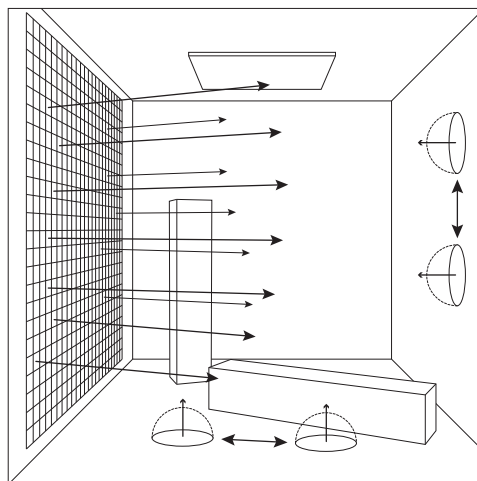


Figura 3.22 Irradiance cache. Segundo paso.



enorme eficacia. Su principal inconveniente, aparte de que requiere un pre-paso, es que al estar basado en muestras aleatorias puede hacer que se pierda detalle en algunos puntos pequeños pero importantes. Y que, en consecuencia aparezcan artefactos, manchas irregulares (*splotches*). Con los algoritmos de fuerza bruta, basados en *distributed ray tracing* y métodos de Montecarlo, el único defecto importante es que se genera ruido. Y en principio el ruido siempre puede reducirse aumentando el número de rayos. Pero con irradiancia pueden aparecer elementos extraños o perderse detalles significativos, aunque también es posible aumentar el número de muestras.

Los métodos de irradiancia son métodos de recolección o de *recolección final* (*final gathering*). La recolección final, como ya hemos visto, es una técnica general, utilizada con diversos sistemas de cálculo, que combina un cálculo inicial relativamente tosco, con un refinamiento posterior pero que tan solo tiene en cuenta un subconjunto de todas las superficies de la escena: el de aquellas superficies que son visibles directa o indirectamente (reflejadas) para el punto de vista activo. Aunque esto supone una reducción importante del número de caras que hay que procesar, los métodos de recolección final a menudo duran bastante más que el cálculo previo, por lo que se han buscado y se siguen buscando mejoras técnicas que reduzcan este tiempo. El método de *irradiancia* o *irradiance caching* es uno de estos métodos. Dado que se basa en el cómputo de una serie mayor o menor de muestras y en la interpolación de los valores resultantes, la mayoría de las técnicas recientes se han centrado en la búsqueda de métodos de selección y formas de interpolación de los valores almacenados en la primera ronda, más eficaces.

La principal ventaja de este método (que es también la principal ventaja de proyección de fotones) es que permite mantener el mapa 3D en que se basa el cálculo, independiente de la geometría de la escena,

sin alterar el modelo, uno de los principales inconvenientes de radiosidad. Y toma medidas muy precisas de la iluminación pero no en cada píxel, por lo que es mucho más eficiente. Si la superficie es regular (plana o con una curvatura uniforme), la interpolación posterior da resultados muy exactos. Si no es así, sobre todo si se producen discontinuidades, puede dar lugar a errores. Este es su principal inconveniente pero que, como ya he dicho, puede reducirse aumentando el número de muestras.

### **Proyección de fotones (*photon mapping*, 1995)**

La proyección de fotones (*photon mapping*) es una técnica que fue desarrollada por Henrik Wann Jensen y presentada públicamente en una serie de artículos a partir de 1995 (Jensen, 1995, 1996). Posteriormente publicó el libro que se referencia en Jensen (2001).

Es una técnica de dos pases que se basa en métodos de Montecarlo para extraer muestras amplias, fiables, de la escena. La idea principal es recoger información sobre la iluminación a partir de un barrido primario desde las luces. Esta información se basa en la generación de una serie de puntos arbitrarios, independientes de la geometría de la escena y que se almacenan en un mapa denominado mapa de fotones (*photon map*). En el primer pase se construye el mapa enviando "fotones" desde las luces. En el segundo pase se hace un *render* de la escena utilizando la información almacenada en el mapa de fotones. Hay que aclarar que estos "fotones" no se parecen mucho a los de los manuales de física pues, entre otras cosas, tienen un radio efectivo que el usuario puede ajustar.

El primer pase incorpora, de hecho, dos estadios: *light emission* y *photon scattering*. En el primer estadio se generan los fotones que se envían a la escena. El número de fotones enviados depende de los valores especificados por el usuario pero también de la intensidad de la luz. La dirección en que



se envían es aleatoria pero queda restringida por el tipo de luz empleada. Una luz direccional emitirá fotones en una dirección principal y una luz volumétrica puede enviar fotones en una dirección preferente si tiene asociado un tipo de distribución determinado. El procesamiento de los fotones es similar a *ray trace*. Al igual que con este método, un fotón encuentra una superficie y almacena la información encontrada. La principal diferencia es que en los métodos de *ray trace* se guarda la información sobre la radiancia del punto de intersección mientras que con las proyecciones de fotones se guarda información sobre el flujo transmitido. Por lo demás, lo que pasa a continuación es similar: el fotón es reflejado o transmitido o ambas cosas, según la información sobre las propiedades materiales del punto de la superficie que se ha encontrado. La dirección de reflexión o transmisión es un vector aleatorio cuya probabilidad depende de la función BRDF y del coseno del ángulo formado por el rayo incidente con la normal. Esto puede implementarse mediante métodos tales como *russian roulette*, que elimina fotones poco importantes y asegura que la energía transmitida se mantiene.

El mapa generado es un mapa de tipo *kd-tree* (una estructura de datos, un árbol *k*, que

subdivide sistemáticamente un espacio de dimensión  $d$  de tal modo que la información guardada en cada nodo puede ser recuperada con gran rapidez). Esta información consiste básicamente en la posición, el color y la dirección incidente, aunque puede ampliarse a otros datos. Así se completa el primer pase.

El segundo pase consiste en la representación de la escena por medio de un procedimiento de *ray trace* tradicional. Cuando el rayo encuentra un punto  $p$  de una superficie, la información almacenada en el mapa de fotones correspondiente al flujo que puede provenir de los fotones vecinos se añade a la información de radiancia (*radiance estimate*), registrada en  $p$ . El número de fotones vecinos que se tienen en cuenta y cuya orientación viene dada por un vector  $d$ , depende de un radio prefijado,  $r$ . Pero muchos de estos fotones se descartan si su producto vectorial  $d \cdot N$  es menor o igual que 0, lo que significa que no apuntan en la dirección adecuada. Los restantes contribuyen con un valor que viene dado por  $\text{Intensidad} \times (d \cdot N) \times \text{factorDifusión}$ . La suma de todas estas contribuciones y de la radiancia estimada para este punto, normalizada, proporciona el color del punto  $p$ . En esta segunda ronda pueden utilizarse diversos controles para ajustar más o menos el cálculo, lo que da mayor flexibilidad a esta

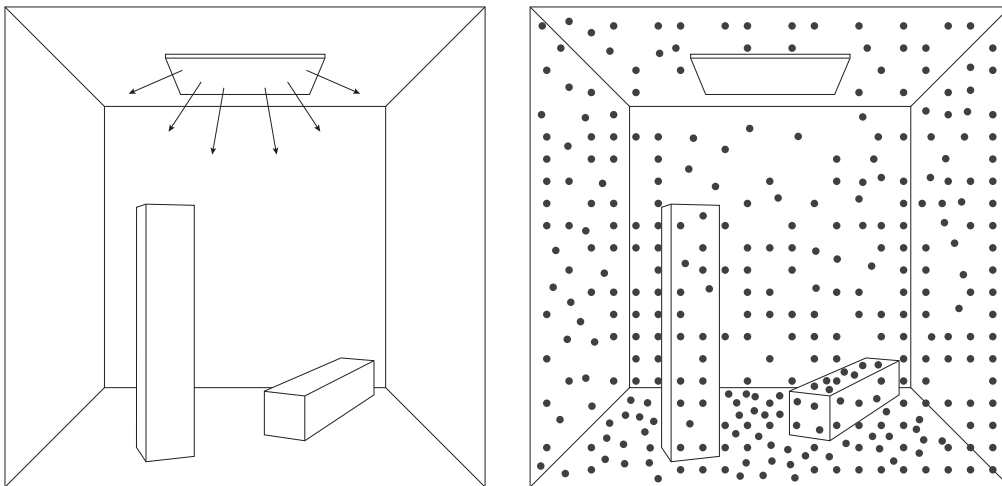


Figura 3.23 Proyección de fotones. Paso 1, desde la luz. Emisión de fotones.

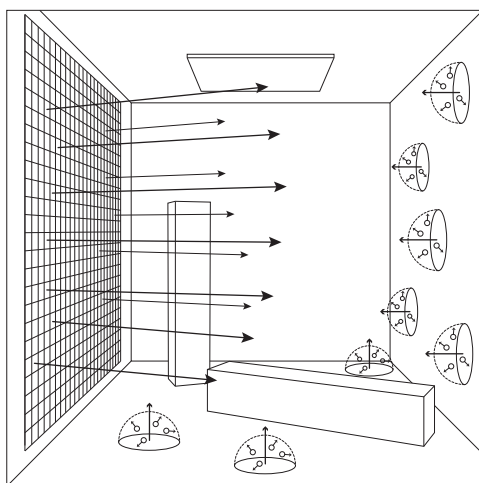


Figura 3.24 Proyección de fotones. Paso 2, desde la cámara. Estimación de la radiancia.

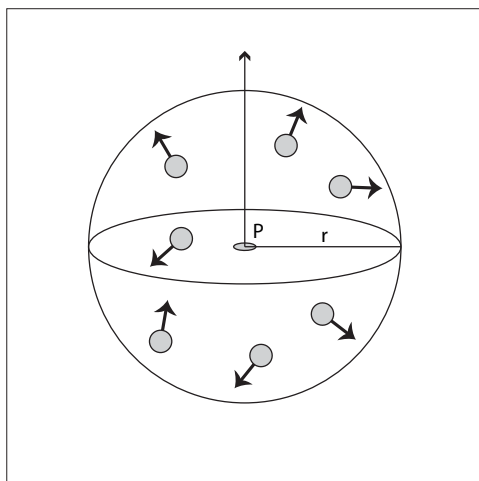


Figura 3.25 Proyección de fotones. A partir del mapa de fotones generado en el paso 1, la iluminación de los fotones vecinos se añade a la información de radiancia recogida en P computando la dirección y la intensidad de los fotones incluidos en una esfera de radio.

técnica y permite jugar con ajustes más toscos para las primeras pruebas y ajustes menos toscos para los resultados finales.

Hay, por tanto tres parámetros principales que afectan a la calidad del resultado y a la duración del cálculo: a) el número de fotones emitidos por la luz, lo que es independiente de la geometría de la escena; b) la precisión del mapa o malla virtual que guarda todas las intersecciones de los fotones con las superficies de la escena; c) la distancia a la que se interpola la información lumínica (el radio de la esfera que se considera en torno al fotón).

El método tiene varias ventajas importantes: puede incorporar con facilidad luces de área, iluminación indirecta, cáusticas, sombras suaves o control de la fuga de color. Procesa por igual geometría arbitraria y polígonos corrientes. Consume poca memoria y pocos recursos y es bastante rápido. Los resultados son una buena aproximación a resultados físicos. Pero se genera ruido que debe ser corregido por métodos complementarios.

El método ha ido evolucionando desde su primera implementación en 1995. A lo largo de la primera década de 2000 se ha ido utilizando progresivamente en combinación con *irradiance caching*.

### Proyección de fotones y cáusticas

Cuando los rayos luminosos que se emiten desde un punto atraviesan una lente o un medio imperfecto, forman un haz que no converge hacia un punto imagen, sino que se dispersa en torno a una curva cuyas propiedades geométricas fueron estudiadas por primera vez en el siglo <sup>xvii</sup> (por Tschirnhausen, en 1682, pero también por Huygens y, posteriormente, por Lagrange en el <sup>xviii</sup> y Quetelet y Cayley en el <sup>xix</sup>, entre otros). La cáustica producida por un anillo, por ejemplo, es una evoluta de envolvente de círculo. Dada una curva  $C$  y un punto fijo  $P$ , se denomina *catacáustica* a la envolvente de los rayos que vienen de  $P$  y se reflejan a partir de la curva  $C$ , y *diacáustica* a la envolvente de los rayos refractados. Los patrones de

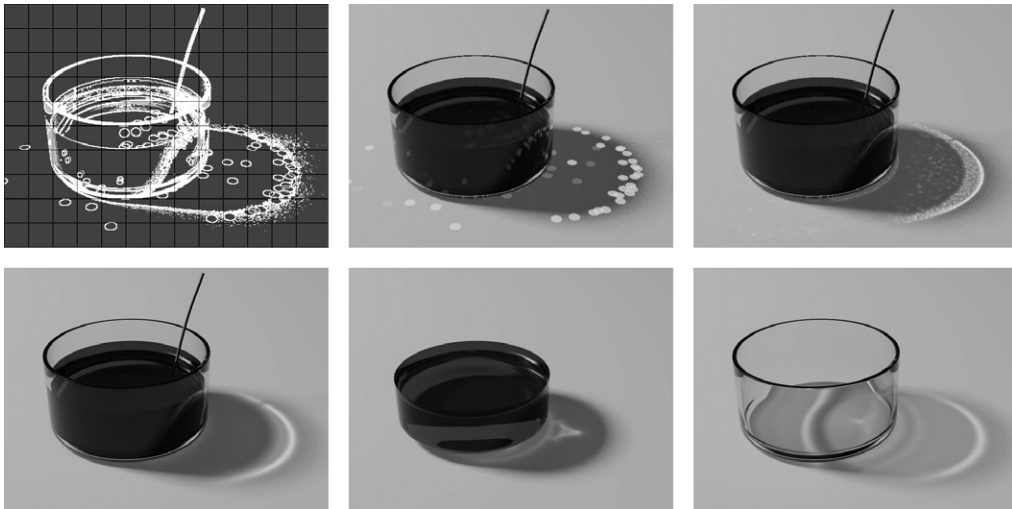


Figura 3.26 Fotones y cáusticas. Las figuras muestran diferentes fases del proceso de simulación. Las dos últimas muestran los dos materiales que intervienen en el proceso (líquido y vaso) aislados.

luz que se forman cuando una fuente de luz atraviesa una botella o un vaso, o los que se forman en el fondo de una piscina o sobre su superficie, son ejemplos característicos de cáusticas.

Las cáusticas no se pueden simular con los métodos corrientes de cálculo de iluminación avanzada como *ray tracing* o radio-sidad. Sin embargo, sí es posible simularlas con métodos específicos, particularmente adecuados para generar estos efectos, como ocurre con los métodos basados en la emisión de fotones.

En arquitectura y diseño este tipo de efectos son poco frecuentes pero hay situaciones en las que pueden mejorar notablemente la calidad de la representación. Así ocurre con escenas en las que aparezcan piscinas, estanques, objetos metálicos o de cristal o, en general, objetos curvos muy reflectantes o transparentes y que sean claramente visibles.

Para simular efectos cáusticos hay que designarlos como tales, hay que especificar qué luces generarán efectos cáusticos y hay que activar los parámetros correspondientes de iluminación avanzada.

En el capítulo sobre técnicas veremos ejemplos concretos de como aplicar esta descripción general.

### Final gather

Para entender mejor el sentido de esta técnica, que puede confundirse con facilidad con las anteriores, resumiré parte de la situación limitándome a tres técnicas: *photon mapping*, *brute force* e *irradiance caching*.

Con *photon mapping* tenemos un algoritmo muy sencillo, razón por la cual ha sido y sigue siendo muy popular entre los programadores. Resumiendo aún más lo ya explicado: a) En el primer pase se envían rayos desde la luz y se crea un mapa de fotones; b) En el segundo pase se envían rayos desde la cámara y estos rayos se encuentran con puntos de una superficie; c) Al encontrar estos puntos, la información que se recoge es, por un lado, la que recibe de la luz, como de costumbre y, por otro lado, la que deriva de los datos almacenados en el mapa de fotones. Si se envían pocos rayos en el primer pase se obtiene un mapa de baja calidad en pocos segundos. Pero el resultado estará lle-





no de manchas desiguales. Si se envían más rayos se obtiene un mapa de calidad media en pocos minutos. Pero el resultado seguirá mostrando manchas aunque más pequeñas. El problema es que para que desaparezcan estas manchas habrá que enviar una cantidad desorbitada de rayos y el resultado puede tardar horas o días.

Con *brute force*, fuerza bruta o Montecarlo simple, tenemos un algoritmo aún más sencillo: a) Por cada píxel de la imagen de salida se envían un montón de rayos hacia la escena en direcciones aleatorias o semialeatorias; b) Cuando uno de los rayos encuentra una superficie vuelve a enviar un gran número de rayos en direcciones aleatorias o semialeatorias; c) Y cuando estos rayos secundarios encuentran otra superficie se repite el proceso hasta un límite especificado por el usuario. El resultado tiende a ser exacto pero los puntos a los que no llegan los rayos, que tendrían que ser infinitos para cubrir toda la escena, aparecen como pequeñas manchas oscuras, como ruido regular. Y para obtener resultados similares a *photon mapping* se necesitan horas en lugar de minutos. Aunque es el método utilizado por algunos programas populares, como Maxwell (y que puede utilizarse por otros programas si se configuran adecuadamente) es, obviamente, poco práctico.

Con *irradiance caching* el procedimiento es similar (aunque se lleva a cabo un pase previo). Se envían rayos hacia la escena y en los puntos de intersección se envían nuevos rayos en todas direcciones. Pero, a diferencia del método de fuerza bruta, no se envían rayos por cada píxel sino por una serie de muestras seleccionadas. Esto quiere decir que habrá puntos de la escena que serán rastreados y otros que serán ignorados. Pero si los puntos ignorados están en el mismo plano que otros rastreados, se pueden interpolar los resultados con un error insignificante. Esto quiere decir que para un gran número de casos (escenas con pavimentos, muros, techos, etc.), el error será inapreciable. Las desventajas son que requiere un paso previo, como en *photon mapping*, y que puede dar

lugar a pérdidas de detalle y a la aparición de manchas por errores de interpolación.

Así que tenemos un método muy rápido, *photon mapping*, pero que da resultados poco presentables y dos métodos más bien lentos pero que pueden dar lugar a ruido o manchas. Y aquí entraría *final gather*, aunque con variantes.

La primera aparición de *final gather* fue en combinación con radiosidad aunque en muchos programas, en la actualidad, este pase se denomina *regathering* o *pixel regathering* para evitar confusiones. En este caso se trataba, como ya hemos visto en el apartado correspondiente, de un pase final que se llevaba a cabo en función de la resolución de la imagen de salida, para suavizar el resultado. Esto se hacía básicamente mediante la inserción de *patches* auxiliares, blancos y perfectamente difusos cuya radiosidad se utilizaba exclusivamente para encontrar un factor de escala con que regularizar, de un modo continuo, los valores finales de los píxeles de la imagen de salida. El principio es, por tanto, similar al del uso posterior aunque el recurso técnico sea muy distinto.

La segunda aparición de *final gather* fue en combinación con *photon mapping*. Y el proceso discurría así:

- a) Se obtiene un *photon map* con el método habitual.
- b) Se inicia el proceso de *rendering* y se envían rayos desde la cámara.
- c) Cuando un rayo encuentra una superficie reenvía algunos rayos desde el punto de intersección para calcular sombras, etc., como en el procedimiento habitual de *ray tracing*. En este estadio de cálculo de la iluminación directa por métodos habituales de iluminación local no se tiene en cuenta el *photon map*.
- d) Desde el punto de intersección se envían más rayos que a su vez, encuentran otros puntos de intersección. En este caso, la información dada por el *photon map* se tiene en cuenta para calcular los valores de iluminación indirecta que se añaden a los de la directa obtenidos en el paso anterior. Ahora bien, esta información adicional se puede





tener en cuenta de dos maneras: como una estimación de la densidad de información en torno a un círculo que rodea al punto (método básico), o enviando nuevos rayos desde este punto hacia puntos vecinos (*final gathering*) y computando los valores de los puntos que encuentre e interpolando los valores para los puntos intermedios.

La tercera alternativa de *final gather* está implicada en lo anterior pues una peculiaridad de esta última alternativa es que, dado que solo se tiene en cuenta los valores de los puntos superficiales que se encuentran, y estos valores dependen tanto de la iluminación que recibe el punto como de su color propio, es posible prescindir del mapa de fotones e incluso prescindir de la luz, pues un rastreo de este tipo estaría computando la iluminación relativa de las superficies que rodean a un punto. De ahí que la siguiente aparición de *final gather* prescindía (o podía prescindir) de la proyección de fotones.

Y aquí aparece la confusión terminológica. Por lo que parece, en algunos programas, el uso de *final gather* se utilizó como sinónimo de *irradiance caching*, sin que hubiera ninguna diferencia apreciable entre ambos usos. Pero en otros contextos (concretamente en las últimas versiones de mental ray) se utilizó como un método que prescindía del precálculo de fotones y computaba directamente la irradiancia que llegaba a un punto superficial enviando rayos desde ese punto a los puntos vecinos.

El procedimiento es similar a *ambient occlusion* (véase el apartado correspondiente más adelante) con la diferencia de que en este caso solo se computan las oclusiones (si hay superficies que obstruirían la luz en la vecindad del punto analizado) mientras que con esta variante de *final gather* se computan también los valores de los puntos de las superficies circundantes, es decir, su radiancia. Dicho de otro modo: las superficies que rodean al punto calculado se consideran como fuentes de iluminación secundaria que se suma a la primaria.

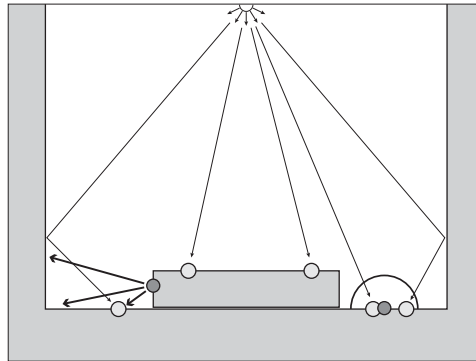


Figura 3.27 Final gather con photon mapping. Las flechas más finas indican trayectorias de fotones y los círculo más claros, los puntos en donde se almacenan en el primer pase. Para computar la luz indirecta en el segundo pase, en puntos indicados por círculos más oscuros, se utiliza una estimación de la densidad por medio de una evaluación en torno a una esfera de un determinado radio (derecha) o final gathering, por medio de rayos enviados desde esos puntos (izquierda, flechas más gruesas).

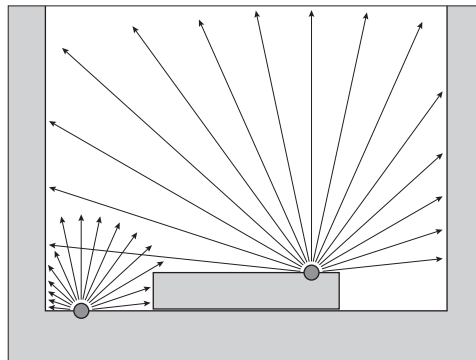


Figura 3.28 Final gather como proceso independiente. Desde los puntos muestreados se emiten rayos en todas direcciones que computan los valores de las superficies de intersección. El proceso puede ser independiente de la iluminación aunque, en general, hay un paso previo en que se computa la iluminación directa.



Una peculiaridad de este último método de *final gathering* es que, dado que los objetos tienen una capacidad intrínseca de actuar como fuentes de luz secundaria, es posible efectuar un cálculo de iluminación con *final gathering* sin que haya luces en la escena (aunque en la gran mayoría de los casos será preferible utilizar al menos una fuente de luz).

Las figuras 3.27 y 3.28 ilustran estas diferencias.

### **Metropolis light transport (1997)**

El método denominado *metropolis light transport* está basado en un algoritmo general, denominado a su vez *Metropolis-Hastings*, que fue presentado por Nicolas Metropolis (Grecia, 1915-1999), junto con otros autores, entre ellos W. Keith Hastings (Toronto, 1930) y que lo desarrollaron en 1953 como un caso especial de la distribución de Boltzman, una distribución de probabilidad de las velocidades de las partículas de un gas, que fue enunciada a finales del siglo XIX.

Se parte de la base de que se cuenta con una distribución de probabilidad en un espacio de estados y se trata de construir una distribución estacionaria. Para conseguirlo, el algoritmo de Metropolis-Hastings diseña una cadena de Markov en dos etapas. En la primera etapa, se propone un nuevo punto por medio de mutaciones aleatorias y, en la segunda etapa, este punto es aceptado o rechazado por una función específica. Si es aceptado, la cadena de Markov se desplaza a ese punto y, si es rechazado, se queda donde estaba. Al seleccionar de modo adecuado la probabilidad de aceptación se llega a construir una cadena de Markov con una distribución estacionaria.

En 1997, Eric Veach y Leonidas J. Guibas presentaron un artículo en Siggraph en el que se describía una variante del método de Montecarlo basada en el algoritmo Metropolis-Hastings. Esta variante partía de construir *paths* desde la cámara hacia la luz utilizando *bidirectional path tracing*. A partir de ahí, se generaban pequeñas modificaciones del *path*.

Por medio de cálculos estadísticos precisos, basados en el algoritmo de Metropolis, se obtenía una distribución de valores adecuada para la imagen final. La principal ventaja de este algoritmo, en relación con métodos generales de *bidirectional path tracing* (véase el apartado correspondiente al comienzo de esta misma sección), es que, una vez que se ha encontrado un recorrido desde la luz al ojo, el algoritmo puede explorar otros recorridos (*paths*) cercanos, que pueden ser de mayor interés pero difíciles de encontrar en una aproximación directa, mediante pequeñas variaciones que reutilizan la información previa, con el mismo número de nodos, que se almacenan en una lista. Una lista que puede ser modificada añadiendo nodos adicionales, si es necesario, para crear nuevos *paths*.

Es un método que no se desvía de los resultados correctos (*unbiased*) y que, en determinados casos, puede hacer que el algoritmo converja hacia una solución correcta de la ecuación de *rendering* con mayor rapidez que otros métodos similares, también basados en *path tracing* o *bidirectional path tracing*.

Su ventaja principal es que es capaz de automatizar un proceso que puede ser fácilmente comprensible para un humano, pero que es difícil que sea interpretado automáticamente por una máquina. Es evidente que en una escena normal en la que, por ejemplo, hay una ventana o una puerta por la que entra mucha luz, los recorridos que se concentran en esas direcciones deben ser explorados de modo prioritario. Pero esta interpretación no puede incorporarse con facilidad a un algoritmo basado en métodos de Montecarlo. El método de Metropolis se adapta de modo natural a la búsqueda de *paths* más significativos que otros.

Esto se consigue, como decía, mediante pequeñas modificaciones de los *paths* iniciales. Estas modificaciones se llevan a cabo por medio de mutaciones. Las mutaciones se llevan a cabo de tal modo que se asegura que la distribución general de muestras, de nodos de *paths*, en la escena, es proporcional a la contribución de estos *paths* a la imagen final. Si un determinado *path* contribuye poco a



esta imagen se desecha en favor de otro más relevante. Pero este nuevo *path* no se crea *ex-novo* sino que se construye a partir de los nodos del *path* previo, con lo que el coste de crear un nuevo *path* es mucho menor que si se generara desde el comienzo. Y este procedimiento potencia la exploración local, que irá favoreciendo los recorridos que contribuyen más a la imagen final a costa de los que contribuyen menos. Una vez que se ha encontrado un *path* significativo, el algoritmo construye otros semejantes con ligeras variantes que aumentan la información en zonas críticas.

Otra manera de explicar lo mismo es que permite crear una distribución que sea proporcional a la *pdf* (*probability distribution function*), lo que resulta ser una alternativa eficiente a su integración (que es prohibitiva). El modo en que las mutaciones son aceptadas o rechazadas, por medio de una función específica que computa una probabilidad de aceptación (*acceptance probability*), garantiza que se mantiene esa proporcionalidad entre la distribución construida y la distribución teórica. El pseudocódigo (dado por Matt Pharr en Jensen *et al.*, 2003, p. 125) sería algo similar a esto:

```
x = x0
for i = 1 to n
  x' = mutate (x)
  a = accept (x, x')
  if (random () < a)
    x = x'
  record (x)
```

Es decir: para  $n$  muestras, aplicar una función de mutación (*mutate*) a cada muestra y una función de aceptación o rechazo (*accept*), y si se cumple la condición de aceptación, almacenar el valor resultante en lugar del previo.

Esto se puede enriquecer por medio de la adición de pesos ponderados al pseudocódigo anterior que favorezcan ciertos valores que se consideran preferentes a otros (*expected values*). El lector interesado en más detalles del algoritmo puede consultar la tesis doctoral de Eric Veach, presentada en 1997, disponible por internet, en que se dan abundantes detalles técnicos de variantes de

optimización. Una versión más reducida pero muy bien sintetizada también puede encontrarse en el artículo citado de Pharr y en otras explicaciones del método que pueden encontrarse por Internet. Aquí, como en los casos anteriores, solo pretendo dar una idea general del procedimiento para que sus aplicaciones técnicas se comprendan mejor.

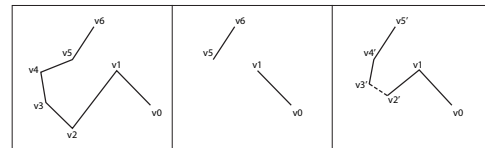


Figura 3.29 Ejemplo esquemático de una mutación bidireccional (adaptado de M. Pharr).

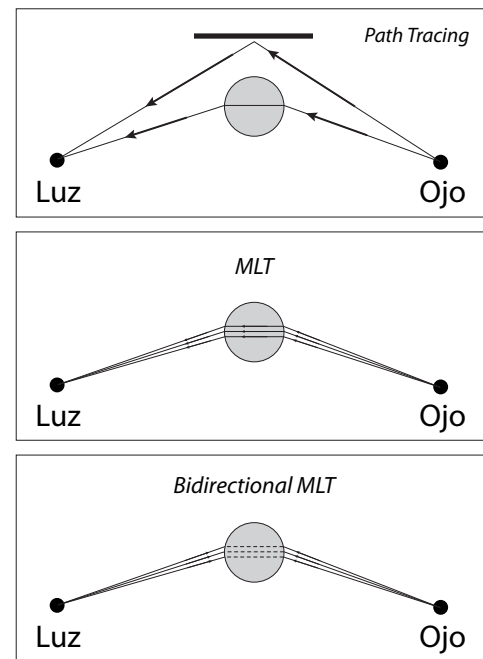


Figura 3.30 Metropolis light transport. Se envían rayos desde la cámara ("ojo") que rastrean la escena hasta encontrar la luz, como en path tracing (arriba). Cuando se encuentra un recorrido de importancia se generan mutaciones, por adición o perturbación de vértices que dan lugar a rayos adicionales cercanos a los primitivos.



El método requiere que se tomen precauciones para no descartar zonas desfavorecidas por el proceso de mutaciones pero que podrían alterar el equilibrio general y una distribución correcta. Las mutaciones deben cumplir unas condiciones generales que se denominan genéricamente condiciones de equilibrio (*balance conditions*).

En su aplicación a la simulación visual de escenarios virtuales, las muestras procesadas por el algoritmo son secuencias de vértices (nodos) del *path* considerado, sobre las superficies de la escena. Es decir, las muestras  $x$  sobre el espacio de estados  $\Omega$  forman secuencias  $v_0 v_1 \dots v_k$ ,  $k \geq 1$  tales que el primer vértice  $v_0$  es una luz y el último,  $v_k$ , es el ojo. A partir de aquí, la estrategia básica es modificar las secuencias  $x$  para dar lugar, por medio de mutaciones, a secuencias  $x'$ , que serán aceptadas o rechazadas según las condiciones de equilibrio estipuladas. Estas mutaciones se efectúan sobre *paths* unidireccionales o bidireccionales.

La figura 3.29 muestra un ejemplo esquemático de las mutaciones que se aplicarían a un recorrido concreto por medio de una mutación bidireccional. A la izquierda se muestra un recorrido de 6 vértices, desde el ojo a la luz, en donde cada vértice del *path* representa una dispersión que tiene lugar en un punto de una superficie de la escena. En el centro, una mutación por la que se han suprimido los segmentos intermedios, lo que da lugar a dos *paths* más cortos que se dirigen a la luz y al ojo. A la derecha se muestra otro esquema derivado del anterior en el que se ha añadido un vértice a cada *path*, vértices cuya posición se ha calculado a partir de la función BRDF de los vértices previos. Para completar el *path* se traza un rayo de sombra entre estos vértices. Si este rayo encuentra un obstáculo la mutación es rechazada y si no, en principio es aceptada.

Al igual que *path tracing*, el algoritmo *metropolis light transport* (MLT) es exacto (*unbiased*). Y, a diferencia del primero, en las pruebas realizadas por los autores, se demostró que era bastante más eficiente y

que permitía reducir significativamente los tiempos de computación. Otra ventaja que hay que citar es que resulta más eficaz en escenas complejas, en las que los algoritmos previos podían no llegar a resolver adecuadamente la representación porque los recorridos no llegaban a puntos recónditos. La flexibilidad del método permite que una vez que se ha encontrado una ruta “importante” es posible explorar otros recorridos cercanos por medio de mutaciones hasta encontrar una variante que llegue a donde los otros recorridos no llegaban. Por añadidura, la reconstrucción no tiene por qué afectar a todo el recorrido sino solo a una parte, por lo que la eficiencia se mantiene.

Pese a todo, es bastante más lento que otros métodos *biased* como *irradiance cache* y *photon mapping* y por eso su utilización comercial es, como veremos más adelante, más restringida (en 2014 está disponible en algunos motores de *render* gratuitos o relativamente poco utilizados, como Maxwell).

### 3.8 Image based lighting (IBL)

#### Fundamentos

*Image based lighting* (IBL) es un sistema de iluminación que prescinde de luces virtuales y, en su lugar, ilumina la escena con un mapa de distribuciones lumínicas. La idea básica es similar a las técnicas de proyección de entorno (*environment mapping*). Sin embargo, los mapas utilizados con IBL están basados en escenarios reales (que pueden coincidir o ser similares al escenario virtual que se va a modelar) y están grabados en formato HDR con lo que la información sobre la distribución lumínica es mucho más precisa. En teoría, la imagen debería abarcar todo el escenario real, es decir, que habría que partir de una fotografía de 360° tomada desde el centro de la escena. En la práctica, este requisito solo se cumple de modo aproximado.



El método tiene importantes ventajas, pues permite una simulación más exacta de la distribución global de la iluminación y, en cierto modo, simplifica el proceso de preparación de la simulación. Pero también implica una serie de complicaciones técnicas que hay que sopesar. En este apartado haré una introducción general a los conceptos generales implicados. En el capítulo 4 se explica con detalle cómo generar los archivos HDR necesarios para este proceso, y en el capítulo 6 se proporcionan ejemplos concretos de las técnicas que hay que utilizar.

IBL tiene relación con otras dos técnicas con las que se puede confundir: *image based modelling*, que consiste en generar modelos geométricos a partir de imágenes, e *image based rendering* (IBR), que consiste en generar modelos visuales aproximados a partir de imágenes. Al igual que en estas dos técnicas, se utilizan imágenes de un escenario determinado para conseguir resultados virtuales. Sin embargo, IBL se basa fundamentalmente en un modelo lumínico, en un mapa de distribuciones lumínicas precisas y su finalidad es conseguir una simulación realista de la iluminación difusa, prescindiendo de luces artificiales.

La principal ventaja de IBL es que se pueden conseguir resultados de gran calidad, más naturales, más semejantes a los efectos que encontramos en las escenas reales en donde las escenas están iluminadas por luces amplias, difusas, con variaciones que dependen de múltiples factores que no es factible simular uno por uno. Otra ventaja adicional es que, en general, no necesitamos crear luces virtuales, orientarlas, ajustarlas, gestionarlas, etc., dado que nos bastará con un mapa principal que simula la distribución global de la luz.

La desventaja principal es que no es sencillo conseguir un mapa adecuado y que esto se agrava porque, en teoría, habría que intentar conseguir un mapa para cada escena, lo que puede ser prohibitivo en muchos casos. Por esta razón, la mayoría de quienes usan este sistema reutilizan los mapas de una es-

cena a otra y, a menudo, combinan un mapa de fondo, visible, con un mapa de entorno que es el que realmente genera la distribución lumínica pero que no se corresponde con el de fondo. Esto resulta poco apreciable en unos casos y más apreciable en otros.

IBL se inició a partir de los trabajos de Greg Ward que, en el programa *Radiance*, ya citado varias veces en los apartados anteriores, incluía recursos para iluminar una escena desde un entorno radiante. Pero el término deriva de la presentación que se hizo en 1998 en Siggraph (véase, en Referencias, Debevec, 1998).

El método presentado originalmente por Debevec incluía algunas variantes que ahora tienen menos sentido en la práctica pero resulta interesante comentar por su relación con desarrollos posteriores. Debevec, además de describir el proceso general, que se resume en los párrafos siguientes, incluye las distinciones siguientes. En primer lugar, parte de la base de que la escena incluirá tres tipos de objetos: la “escena distante” (*distant scene*), la “escena local” (*local scene*) y los “objetos sintéticos” (*synthetic objects*). La escena “distante” es el mapa HDR que incorpora la información de luminancias, pero no incluye información de reflectancia y no es procesado por el sistema. Los “objetos sintéticos” son los modelos virtuales que hemos incorporado a la escena junto con sus respectivos modelos de reflectancia. La escena “local” es el soporte de los objetos sintéticos, que será afectada por sus sombras y reflejos y que, en la práctica, a menudo se reduce a un simple plano.

Esto plantea el problema de que este plano, que incorpora también un modelo de reflectancia, debe fundirse adecuadamente con el fondo, la escena “lejana”. Sin embargo, aunque los materiales con que se simula la escena local sean iguales que los de la lejana, se producen inconsistencias en la iluminación debido a que esta superficie no queda iluminada igual que el fondo. Para corregir este problema, Debevec introduce un cálculo complejo, que denomina *differential rende-*

*ring*, y que básicamente consisten en calcular el error o diferencia que aparecen entre la imagen de la escena local (por ejemplo, el plano en que se apoyan los objetos) tal como aparece en la imagen del fondo y tal como aparece cuando se procesa. Es decir, la diferencia entre las características de reflectancia de la escena local “real” (fotografiada) y la escena local “virtual” (generada). Esta diferencia se incorpora a un algoritmo, introducido en el programa *Radiance*, que se utilizó originalmente para la simulación lo que da como resultado una fusión perfecta. Como veremos más adelante, esta técnica, difícil de incorporar a un programa de *rendering*, se puede obviar si se utilizan recursos tales como *shaders* que generan sombras y reflejos sobre una plano invisible (por ejemplo, el *shader matte/shadow/reflection* de mental ray).

La figura 3.31 muestra un esquema adaptado del artículo original de Debevec.

### Procedimiento general

El procedimiento, al margen de estos matices, consta por consiguientes de los pasos siguientes, que se describirán con más detalle, como ya he dicho, en los capítulos correspondientes, presentando las técnicas específicas mediante ejemplos concretos.

a) Se crea un mapa que capte la distribución de una escena real o virtual (preferentemente real) desde todas las direcciones que

rodean a una zona determinada. Para crear este mapa hay diversos métodos que se analizarán en el capítulo 4. Podemos resumirlos en cinco alternativas: 1) utilizar una bola especular, situada en una posición adecuada en el escenario real o en un escenario similar al que se va a simular, y grabar varias tomas a diferentes exposiciones para, a partir de estas diferentes tomas, generar un único archivo HDR; 2) captar diferentes tomas rotando la cámara y luego componer estas tomas en un panorama HDR con programas especiales (programas de *stitching*); 3) utilizar una cámara con un gran angular para captar estas diferentes tomas, lo que reduce el número de tomas, o una cámara especial (muy cara) capaz de grabar directamente archivos HDR mientras rota en torno a la escena; 4) crear archivos HDR directamente a partir de escenarios virtuales que simulen adecuadamente las condiciones de iluminación ambientales; 5) adquirir archivos HDR de terceros que representen escenarios similares a los que queremos simular.

Hay que observar que algunos de estos métodos pueden no cumplir plenamente las dos condiciones ideales que deberían cumplir estos mapas, sobre todo la primera: ser omnidireccionales, es decir, presentar un valor lumínico para cada dirección del entorno y, en segundo lugar, incorporar valores, para cada píxel, que sean directamente proporcionales a la cantidad de luz que tendría la escena real. Pero pueden ser suficientes en la práctica real.

b) Este mapa se asigna al entorno de la escena como una proyección esférica, o bien a una luz celeste (*sky light*) que ilumine la escena desde un hemisferio virtual, o bien se asigna, como material, a un objeto hemisférico que abarque toda la escena. En cualquier caso, en principio, la escena no contará con otra fuente de luz que el mapa asignado al hemisferio que la envuelve, de uno u otro modo, tal como se muestra en la figura 3.32. Hay que tener en cuenta que aparecerán complicaciones técnicas que ya analizaremos, pues la resolución y el enfoque requeridos para un

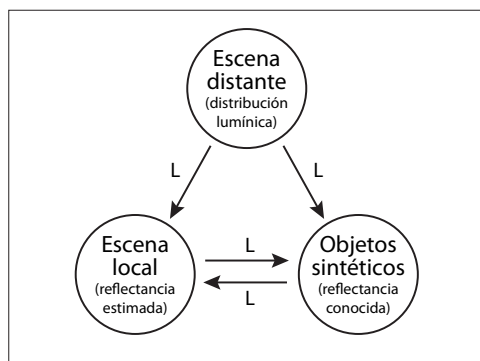


Figura 3.31 Esquema del cálculo básico de IBL (adaptado de Debevec, 1998).





mapa de fondo, para un mapa de iluminación y para los reflejos no tienen porque ser los mismos.

c) Este escenario se procesa por un sistema adecuado de simulación avanzada, un sistema de iluminación global (GI) capaz de computar la interacción entre objetos. El proceso habitual será alguna variante de métodos de Montecarlo que enviará rayos a la escena. Rayos que, al encontrarse con un punto de una superficie, reenvían rayos en todas direcciones para computar la iluminación proveniente de otras superficies. Si el rayo no encuentra otras superficies leerá directamente la información dada por el mapa IBL, tal como también se muestra en la figura 3.32. Si encuentra otra superficie en su camino computará los valores de reflectancia de esta superficie y, si esta superficie tiene cierto grado de especularidad, leerá indirectamente los valores dados por el mapa IBL a través de los rayos secundarios que se derivarán de la reflexión especular. También puede procesarse la escena con programas especializados sencillos especializados en IBL. Prácticamente cualquier sistema de iluminación global puede procesar una escena cuya única complicación importante es el uso de un mapa adecuado.

Hay algunos aspectos, avanzados en el punto (b), que conviene subrayar. El procesamiento de un mapa de alta resolución como fuente de luz supondría una carga importante para el sistema y daría lugar a artefactos (*aliasing*), como ya he dicho antes. Para evitar esto, muchos programas procesan el mapa con un filtro de convolución lo que equivale a tomar un número determinado de muestras e interpoladas. Esto equivale a substituir el mapa por un mapa de irradiancia o a aplicar al mapa un filtro adecuado, por lo general un filtro gaussiano. Parece obvio que, en determinados casos, será más sencillo cargar directamente un mapa ya filtrado.

Sin embargo, esto da lugar a problemas con los fondos o los reflejos que pueden requerir lo contrario: un mapa preciso que permita ver claramente el fondo o computar

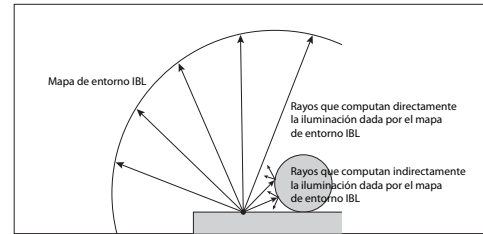


Figura 3.32 IBL. Proceso de cálculo en un escenario con un mapa de entorno.

adecuadamente los reflejos en zonas especulares. En el caso de los reflejos, también se puede solucionar por medio de *mipmaps* que permitan utilizar submapas de baja resolución para reflejos difusos y submapas de alta resolución para reflejos nítidos.

De todo esto deriva un flujo de trabajo que veremos en la práctica y que a menudo consiste en utilizar diferentes mapas HDR, con diferentes características, para la iluminación, para el fondo y para los reflejos.

Otro problema que habrá que tener en cuenta, y que también se discutirá más adelante, es el tipo de mapa de entorno a utilizar. La imagen captada por una bola especular se adapta directamente a un mapa esférico, pero presenta grandes distorsiones en los bordes que pueden dar lugar a artefactos. Una alternativa relativamente corriente es utilizar un mapa cúbico, pero esto obliga a reprocesar los mapas y tampoco es una solución ideal. Estas alternativas, y otras, también se discutirán más adelante.

IBL funciona particularmente bien si se combina con *ambient occlusion*. Este es el método que utiliza el RenderMan de Pixar. Pero no está integrado en un mismo procedimiento en otros programas por lo que habrá que combinarlos manualmente. Por añadidura, RenderMan utiliza un procedimiento más efectivo que las variantes de *ray tracing* habituales, un procedimiento denominado *point-based approximation* que se basa en crear una nube de puntos que abarca toda la escena y en la que cada punto (en realidad, un pequeño disco con una orientación concreta) almacena información relevante



para el cálculo de iluminación. No lo describo porque es un método muy específico que no está disponible en los programas de simulación corrientes. El lector interesado puede visitar la página <http://renderman.pixar.com/view/pointBased-occlusion-and-color-bleeding>.

### 3.9 Ambient occlusion

Los métodos de cálculo de la oclusión ambiental (*ambient occlusion*) se deben a Industrial Light & Magic, la compañía fundada por George Lucas. Se desarrollaron por un equipo dirigido por Hayden Landis, se utilizaron en películas como *Pearl Harbour* y *Stuart Little 2* y los resultados se presentaron en Siggraph, en 2002 (véase Landis, 2002; Bredow, 2002). Con todo, ideas similares habían sido avanzadas en 1998 por Zhukov que introdujo el concepto de *obscurances* (véase Zhukov, 1998). Partía de la base de que se podría lograr una representación de interés evaluando el grado de ocultación (*hiddenness*) u oclusión de un punto al tomar en consideración los objetos que le rodeaban. El valor de “oscuran-

cia” (*obscurance value*) tendría un rango de 0 (completamente oculto) a 1 (completamente no oculto). Ideas muy similares a las desarrolladas posteriormente por el equipo de Landis aunque con diferencias técnicas importantes. En cualquier caso, es un método que prescinde por completo de cualquier fuente de iluminación o, mejor dicho, presupone una iluminación global uniforme que abarcaría toda la escena.

La idea básica es computar el modo en que la distribución de esta luz global quedaría atenuada por objetos de la escena que la bloquearían en mayor o menor grado. Esto se traduce en la utilización de diversos métodos para computar lo que “ve” un punto al inspeccionar su entorno. El modo más sencillo de hacerlo es mediante un mecanismo de *ray tracing* que envía rayos desde un punto de una superficie y computa si estos rayos encuentran algún obstáculo en su camino hacia la luz o hacia un mapa de entorno que recubre la escena.

Como veremos, hay otros métodos, pero este es el método principal. Implica un paso previo por el que se obtiene la información requerida para una serie de muestras (triángulos) de la superficie, tanto de la oclusión como de la dirección de donde llega la luz visible. Esta información se utiliza, durante el proceso de *rendering*, para modificar el color de las superficies.

Las figuras 3.33 y 3.34, muestran los elementos básicos de este método, así como las dos variables principales que intervienen. Matemáticamente, la oclusión  $A_p$  en un punto  $p$ , se obtendría resolviendo la ecuación siguiente:

$$A_p = \frac{1}{\pi} \int_{\Omega} V_{pw}(Nw) dw$$

donde  $V_{pw}$  es la función de visibilidad del punto  $p$  en la dirección  $w$ ,  $N$  es la normal a la superficie en el punto  $p$ , y  $dw$  es el ángulo sólido infinitesimal considerado para la variable de integración  $w$ . La función  $V_{pw}$  está definida de tal modo que su valor es 0 si hay oclusión en la dirección  $w$  y su valor es 1 en caso contrario.

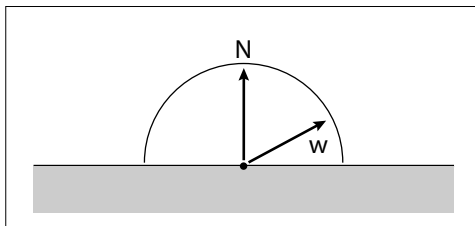


Figura 3.33 Ambient occlusion. Esquema básico.

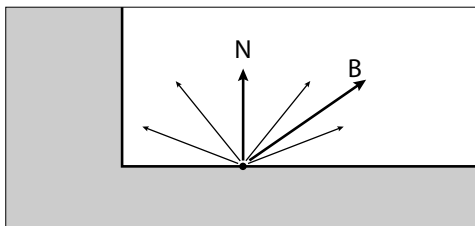


Figura 3.34 Ambient occlusion. Vector derivado de dirección media sin oclusión.



A menudo se utiliza también un vector adicional denominado *bent normal* (normal inclinada), denominado  $N_b$  o  $B$ , que apunta en la dirección media en la cual no hay oclusión. Este vector puede utilizarse para computar la radiancia incidente que proviene de un mapa de entorno (por ejemplo, cuando se utiliza IBL), lo que permite combinar estos dos tipos de cálculo, tal como he indicado en la sección anterior con respecto a Pixar. Hay que tener en cuenta que, dado que este vector se computa a partir de valores medios, se pueden producir errores. Por ejemplo, en el caso de que a ambos lados del hemisferio no hubiera oclusiones, el sistema podría derivar un valor medio que apuntará hacia arriba. Pero en esta dirección podría haber un obstáculo no detectado.

Como he dicho antes, hay otros métodos. Además del citado, de Pixar, hay métodos de fuerza bruta que sacan partido de la extraordinaria potencia de las tarjetas gráficas actuales y de su capacidad para almacenar mapas de textura en *buffers* especializados. Uno de estos métodos es, en lugar de enviar rayos desde cada punto de las superficies de la escena, invertir el planteamiento y rodear la escena por múltiples luces (creadas virtualmente durante el procesamiento propio de este método) distribuidas de modo uniforme sobre una esfera virtual o, si hay un plano base principal, por un hemisferio virtual. Estas luces computan las sombras por medio de mapas de sombra (*shadow maps*). La oclusión en un punto se calcula como la media dada por los mapas de sombra generados por cada una de las luces. Esto quiere decir que si hay  $n$  luces habrá que hacer  $n$  renders, desplazando el cálculo de sombras a la luz siguiente en cada caso. Los resultados se acumulan en un mapa en blanco y negro que se procesa para unificarlos en un mapa único que representará las sombras difusas creadas por la esfera virtual, si emitiera luz.

A pesar de que se requieren entre 128 y 1.024 luces para obtener buenos resultados, las pruebas que se han hecho demuestran que con una tarjeta gráfica relativamente co-

rriente los resultados son más rápidos que con *ray tracing*. Hay, con todo, varios problemas técnicos, artefactos que aparecen si la distribución de las luces no es adecuada, etc., que pasaré por alto. También debe tenerse en cuenta que se requieren dos pasos, uno para calcular y almacenar los resultados y otro para hacer el cálculo final de *rendering*, por lo que si hay  $n$  luces habrá que hacer  $2n$  pasos. Pero también para esto hay recursos, trucos técnicos que reducen este coste. Véase el artículo sobre este tema que se encuentra en las GPU Gems de Nvidia (cap. 17). En este artículo se menciona que *ambient occlusion* puede verse técnicamente en el contexto de las técnicas de iluminación basadas en *spherical harmonics*, como una simplificación extrema que utilizaría tan solo la primera *spherical harmonic* para representar la reflexión.

Con todo, el método más utilizado es el que he descrito en primer lugar, que da resultados bastante satisfactorios de un modo muy simple y con notable rapidez. Lo habitual es combinar los resultados obtenidos con otros sistemas de cálculo, tal como veremos en los ejemplos concretos que se incluyen en los capítulos que siguen.

### 3.10 Otros métodos

#### *Importons*

El uso de “funciones de importancia” (*importance functions*) se remonta a simulaciones de flujos de neutrones tras la 2ª Guerra Mundial, y sus primeras aplicaciones al campo de la simulación de iluminación se incorporaron a algunos algoritmos de aceleración de *ray tracing*, publicados en artículos destacados de esta época, debidos a algunos de los principales investigadores como Arvo, Cook, Hall, Greenberg o Ward, entre 1983 y 1991, artículos en los que de algún modo estaban implicados estos métodos. Entre los primeros artículos en que aparece esta denominación aplicada expresamente a un algoritmo de iluminación global, de radiosidad



concretamente, se pueden mencionar el de B. Smits, J. Arvo y D. Salesin en 1992 ("An Importance-Driven Radiosity Algorithm") o el publicado por Pattanaik y Mudur en 1993 ("The Potential Equation and Importance in Illumination Computations", citado en las referencias). Pero hay muchos artículos posteriores que incorporan estas técnicas, por lo que no cabe asignar una fecha concreta a sus aplicaciones iniciales. Así, artículos dedicados a optimización de *path tracing* (véase, por ejemplo, Jensen, 1995) a optimización de *bidirectional path tracing* (véase, por ejemplo, Lafortune y Willens, 1993) o a optimización de proyección de fotones (varios artículos en torno a 2000). Las diversas aportaciones a estos temas están bien resumidas en un artículo de Per Christensen (2003, véanse las referencias).

El detalle de estos métodos se basa en técnicas matemáticas para extraer ecuaciones asociadas a integrales generales, que permitan filtrar (por medio de transformaciones de funciones) los valores más significativos descartando los menos significativos. En el caso de ecuaciones de cálculo de iluminación, las ecuaciones que sintetizan el transporte de la luz tienen términos básicos que se pueden asociar con regiones de especial interés, tales como el punto de vista, el plano de la imagen o las superficies de la escena que son visibles desde la cámara. Estas regiones de interés pueden ser incorporadas al análisis o, para decirlo de un modo más gráfico, pueden ser "transportadas" de modo similar a como se transporta la luz. Christensen lo ilustra de un modo muy claro: si apagáramos las luces y "encendiéramos" una luz que coincidiera con una región de interés tal como el plano de la imagen, y dejáramos que esta luz se expandiera y rebotara en la escena hasta llegar al equilibrio, la contribución a la imagen final de estas partes a las que llega esta luz sería proporcional a la cantidad de luz que llega a ellas y, en consecuencia, a su importancia con respecto a esta región. Si el cálculo tiene en cuenta estos diferentes grados de importancia y se refina cuando im-

porta pero se simplifica cuando no, es obvio que la ganancia en eficiencia será considerable.

En relación con la radiancia, la *importance* puede definirse como la fracción de la radiancia que contribuye, directa o indirectamente, a la región de interés. Esta fracción puede normalizarse de 0 a 1 de modo que si la contribución es máxima su contribución es 1. La *importance* se representa a menudo como una función de salida, exitancia  $W_e(x, w)$ , del punto  $x$  en la dirección  $w$  o como una función de entrada, incidente,  $W_i(w, y)$  que viene de la dirección  $w$  hacia el punto  $y$ . Si los puntos  $x$  e  $y$  son mutuamente visibles tendremos que  $W_e(x, w) = W_i(w, y)$ .

El procedimiento general es el siguiente. Primero se emiten partículas especiales, *importance particles* o *importons*, como se les ha denominado, desde la cámara hacia la escena. Estos *importons* quedan restringidos al frustrum, la pirámide de visión. Cada vez que se encuentran con una superficie se registra su posición y su valor. En cada rebote, la partícula modifica su valor en función del rebote anterior. Podemos imaginarnos que son inicialmente blancas pero que al rebotar en una superficie de color rojo, pierden luminosidad y adquieren una tonalidad rojiza, según el grado de reflectancia de la superficie. Después de que se han trazado todas las partículas, su importancia es estimada en función de la densidad de cada zona.

A esto le puede seguir un trazado de fotones. Pero en este caso, los fotones almacenados utilizan una función de probabilidad basada en los resultados anteriores, en los *importons*, para ser almacenados o descartados. Si se descartan, su pérdida se convierte en ganancia para los almacenados con lo que la energía global se conserva.

En las aplicaciones disponibles (principalmente con mental ray) se pueden utilizar de varios modos.

Uno de estos modos es en combinación con mapas de fotones. En estos casos, los



fotones cercanos a un *importon* se mantienen y los que no, se descartan (más exactamente: su energía se redistribuye entre los fotones restantes). Esto permite aligerar considerablemente el peso de los mapas de fotones sin pérdida apreciable de precisión. Y con esto se añade otra fase al cálculo pues los *importons* se disparan antes que los fotones. Las aplicaciones que usan esta técnica permiten configurar el cálculo de modo que se dispare una cantidad adecuada de *importons* según el tamaño de la imagen, la densidad deseada, etc., de modo bastante similar a las configuraciones características de fotones. Los dos parámetros principales son *density* y *trace depth*.

### **Irradiance particles**

Otro modo es utilizar *importons* sin fotones con la ayuda de *irradiance particles*, otra nueva técnica que parte de la base de que ya existe un mapa de *importons* en la escena generado antes de la fase de *rendering*. Este mapa guarda, en cada punto, información sobre la iluminación directa y la posible iluminación indirecta que recibe. El término *irradiancia* se refiere precisamente a este tipo de información. La iluminación indirecta puede ser más o menos precisa según el número de pases previos al cómputo definitivo del mapa y solo los valores “importantes” se registran. Durante la fase de *rendering* se utilizan *irradiance particles* para estimar la irradiancia de cada punto que será representado. Los parámetros de configuración son similares a los de FG de mental ray: número de rayos, número de pases, valor de interpolación y rayos de entorno. Por ejemplo: 680/2/32/480.

El parámetro de configuración principal en estos casos es *trace depth* que especifica el número de rebotes (*bounces*) de los *importons*. Se requieren valores altos para que toda la escena sea barrida adecuadamente. Se puede empezar con un valor de densidad de 0,2 y un valor de *depth* de 2 e ir subiendo hasta valores tales como 0,5 y 4.

Tanto *importons* como *irradiance particles* son técnicas de optimización que buscan reducir de modo significativo el tiempo de cálculo. Dada la rapidez con que sigue aumentando la potencia de los ordenadores, está por ver si las complicaciones técnicas de estos métodos (que tampoco lo son tanto) compensarán a los usuarios por una ganancia de tiempo que puede ser imperceptible en muchos casos. Pero como ha ocurrido con otras técnicas, ahora obsoletas, tiene interés mencionarlás pues quizás permiten intuir la dirección en que pueden moverse los métodos de cálculo en un futuro inmediato.

### **Brick maps. Point clouds. Point based GI**

La aplicación *RenderMan* utiliza dos métodos propios, pero que, aparte de su notable interés, tienen similitudes con los utilizados por otros programas: *brick maps* y *point clouds*. Estos métodos se introdujeron a partir de 2006 y se utilizaron parcialmente en la película *Ratatouille* (2007) y extensamente en *Wall-E* (2008).

*Brick maps*, el método más específico de los citados, se basa en los pasos siguientes:

a) Se calcula la iluminación directa en un *render* previo y, durante el cálculo, la aplicación genera una nube de puntos (*point cloud*) en la que cada punto de la nube almacena la iluminación directa que recibe la escena. Este paso se denomina *baking the direct illumination*.

b) La aplicación convierte esta nube de puntos en un *3D brick map*, es decir, en algo muy similar a un mapa procedural 3D. Este mapa es independiente de la vista de cámara.

c) Por último, en el *render* final, se calcula la imagen de salida de modo que, para cada punto del que se necesita conocer la iluminación (indirecta o emitida), la aplicación dispara rayos inversos hacia el *3D brick map* y registra el color del punto correspondiente. Para reducir el cálculo de millones de puntos, se utilizan métodos de



optimización que reducen drásticamente el número de rayos necesarios para inspeccionar el *brick map*. En cierto modo, el método es similar al de REYES, utilizado por RenderMan desde sus inicios, que también divide las superficies en micropolígonos.

*Point clouds* (una denominación genérica que puede tener diferentes sentidos en otros contextos) es un método muy similar que se basa en los pasos siguientes:

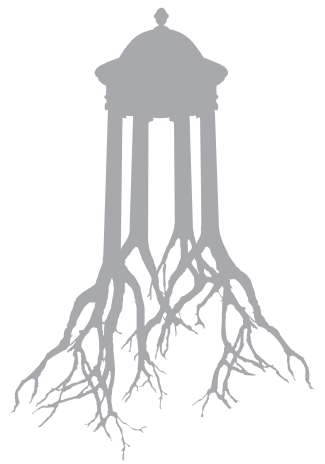
a) Como en el caso anterior, se calcula la iluminación directa en un *render* previo y durante el cálculo la aplicación genera una nube de puntos (*point cloud*) en la que cada punto de la nube almacena la iluminación directa. Pero la diferencia principal con el método anterior es que no se genera un *brick map*.

b) Se pasa directamente al *render* final, calculando la imagen de salida mediante el procesamiento de puntos por medio de árboles octales (*octrees*). Los puntos cercanos se evalúan completamente pero los puntos lejanos se agupan en una lectura única. La idea básica es que cada punto registra todo lo que tiene alrededor, computando por tanto la iluminación indirecta que recibe de sus vecinos, como si cada punto fuera una pequeña cámara en ojo de pez que computase un *mini render* por medio de un proceso clásico de rasterización, pero utilizando una estructura de árboles octales para acelerar el proceso y evitando computar todos los puntos de la nube. De este modo, se evita el uso de *ray tracing* y se simplifica el cálculo. También se utiliza un sistema de *cache* para leer puntos a medida que se necesitan, así como otros métodos de optimización.

*Point based GI* es otra alternativa a *ray tracing* que utiliza métodos similares a los anteriores. La idea básica es representar la escena por medio de una aproximación, de la toma de una serie de muestras a partir de la cual se construye una nube de “puntos” o pequeños discos, a veces denominados *surfels*, y computarla por medio de tres pases: en el primero, se genera una nube de puntos

con micropolígonos iluminados directamente, en el segundo, se computan  $n-1$  puntos de la nube (o menos o ninguno según los casos), y, en el tercero, se computa la iluminación indirecta dada por la nube de puntos. Uno de sus pioneros principales es Christensen, que la utilizó en la película *Pirates of the Caribbean* y *Surf's Up* y luego en muchas otras películas. Tiene similitudes con *irradiance* en la medida en que se toman muestras aisladas como base para el cálculo.

La estructura inicial de la nube es importante pues afecta a la velocidad con que puede recorrerse durante la fase de cálculo. Es una estructura jerárquica, por lo general un árbol octal (*octree*) o alguna variante similar, en donde cada nivel en la jerarquía agrupa de un modo más o menos simplificado los niveles inferiores que almacenan la geometría y la radiancia de los puntos. Para simplificar el proceso, que implicaría recorrer toda la estructura, hay varios métodos de optimización. En el apartado correspondiente sobre GPU, en el capítulo 4, ampliaré algo más estos temas.







## **TÉCNICAS. ANÁLISIS DE CASOS**

→ 4



# Recursos básicos y complementarios

## 4.1 Recursos básicos

Los métodos de simulación, tanto de materiales como de iluminación, dependen cada vez más de recursos que se van incorporando progresivamente a las tarjetas gráficas cuya potencia, en ciertos aspectos, ha superado a la de los ordenadores corrientes. La situación es compleja pues hay ciertos procesos muy complicados pero que se realizan a gran velocidad debido a que están incorporados al procesador principal de estas tarjetas, mientras que otros permanecen ligados al software y al procesador de la placa base.

Es conveniente comprender mínimamente cuáles son estos procesos y cuál es su posible nivel de integración para sacar el mejor partido de los programas de simulación.

Como ocurre cada vez más en cualquier trabajo que se basa en procesamiento digital, resulta difícil precisar qué quiere decir esto de comprender “mínimamente”. El criterio que he seguido es desarrollar este tema con cierta amplitud en el libro sobre simulación visual de materiales, en un capítulo relativamente extenso sobre este tema, pues la simulación de texturas, por ejemplo, es un aspecto muy importante de esta integración.

Para evitar repeticiones, en esta sección me limitaré a resumir lo principal. El lector más interesado en estos temas puede saltarse directamente este resumen y dirigirse al capítulo mencionado del libro sobre simulación de materiales y a las referencias que se dan en aquel capítulo.

### Las tuberías de la representación

#### Tarjetas gráficas. La GPU

El procesamiento de imágenes introdujo, desde la aparición de los ordenadores, una

carga importante sobre la memoria y la velocidad de cálculo, lo que llevó a introducir adaptadores especiales que se hicieran cargo de este proceso. En 1987 se introdujo (por IBM) lo que sigue siendo un estándar de referencia, la tarjeta gráfica básica, VGA (*video graphics array*) que tenía una resolución de 640 x 480 píxeles y una profundidad de 16 colores. Los fabricantes de chips especializados comenzaron a desarrollar alternativas a este estándar y desarrollaron unidades independientes con mayor capacidad.

A mediados de la década de 1990 se comenzó a utilizar progresivamente el término GPU (*graphics processing unit*) para describir a estas unidades independientes que se conectaban por medio de un conector especial a la placa base. Por esas mismas fechas comenzó una carrera industrial que, en pocos años, dejó a dos compañías que abarcaban, y abarcan, la práctica totalidad del mercado de tarjetas gráficas profesionales: NVIDIA (fundada en 1993), y ATI (Array Technologies Incorporated, fundada en 1991) actualmente AMD (Advanced Micro Devices) tras su absorción por esta compañía en 2006.

A medida que avanzaba esta independización de las tarjetas gráficas con respecto a las unidades centrales, crecían las dificultades para programarlas, pues había que dedicar mucho tiempo a elaborar códigos de transición, códigos que adaptaran el funcionamiento interno de estos dispositivos independientes al funcionamiento de los dispositivos receptores. Esto llevó al desarrollo de aplicaciones genéricas que actuaban como mediadoras entre ambos extremos. Así surgió la primera, OpenGL, en 1992. Y, desde Microsoft, y por tanto mucho más ligada a un sistema operativo concreto, Windows, surgió



DirectX en 1995. Durante muchos años la primera se destacó como más fiable y genérica. Pero poco a poco la segunda empezó a ganarle terreno. La historia y las características de esta competición y, sobre todo, de sus implicaciones técnicas, la retomaré en el apartado siguiente.

Las nuevas tarjetas gráficas que continuaron su evolución durante estos años fueron ampliando progresivamente sus capacidades. Se añadieron funciones matemáticas avanzadas, procesos básicos de asignación de materiales y de procesamiento de texturas, de iluminación básica que permitieran interactuar directamente con los modelos virtuales, etc. La serie de transformaciones implicadas en el proceso se fue estandarizando, lo que facilitó también la colaboración de diferentes programadores para introducir mejoras en diferentes puntos de este proceso complejo, que también retomaré en el apartado correspondiente a la descripción de las “tuberías de la representación”. Dentro de este esquema adquirirían una importancia creciente los *shaders* y los estadios conocidos como *vertex shaders* y *pixel shaders*, a los cuales también volveremos.

Las especificaciones que se pueden encontrar (en 2014) en un catálogo de tarjetas gráficas incluyen varias categorías que se pueden agrupar en tres o cuatro grupos principales que resumo a continuación, recordando de nuevo que se puede encontrar una descripción bastante más amplia en el libro sobre simulación de materiales.

a) El componente principal o *motor principal* de la tarjeta es la GPU (*graphics processing unit*), un chip de mayor tamaño (de 2 a 6 cm aproximadamente), que ocupa una posición central y que suele incorporar un ventilador para disipar el calor que genera su funcionamiento. El dato característico suele ser la velocidad, pero este valor, que suele estar en torno a los 900 MHz en la tarjetas de gama media o alta, muy inferior al de la CPU, es equívoco pues la gran potencia de las tarjetas gráficas no deriva de su velocidad sino de su estructura, que potencia considerablemente

el paralelismo y que saca partido de anchos de banda de comunicación mucho mayores. Otro tanto cabría decir del número de transistores, que puede oscilar entre algo menos de 1.000 millones y algo más de 7.000, pero cuyo número, mayor o menor, es relativamente irrelevante con respecto a la estructura. De ahí que suele ser más revelador fijarse en datos tales como el número de operaciones de transferencia por segundo que se mide en GT/s (*giga transfers* por segundo). Las GPU de gama baja o media suelen alcanzar (2014) valores de 10 a 100 GT/s para el procesamiento de cosas tales como el número de unidades de textura que se proyectan sobre la pantalla por segundo, y las de gama alta alcanzan valores superiores a los 200 GT/s. Otro valor revelador es el número de operaciones por segundo que se mide en GFLOPS (109 FLOPS, *floating point operations per second*). A veces se encuentra la denominación GFLOPS (FMA) que indica que las operaciones de multiplicación y suma están integradas en un solo ciclo (*fused multiply add*). Las GPU de gama baja suelen estar por debajo de los 500 GFLOPS, las de gama media en torno a los 2.000 y las de gama alta por encima de los 3.000.

b) Otro componente principal es la *memoria*. Aunque en las tarjetas antiguas la memoria se relacionaba directamente con la resolución mediante un cálculo simple, en las modernas este cálculo es poco relevante. La mayoría de las tarjetas actuales, tanto de gama baja/media como de gama alta, alcanzan una resolución de 2.560 x 1.600 píxeles. Esto supone procesar algo más de 4 millones de píxeles ( $2.560 \times 1.600 = 4.096.000$ ) lo que, para 32 bits por píxel, supone 15,6 mb. Pero esta resolución se almacena en múltiples lugares: un *back buffer* que substituye con rapidez al *frame buffer* que se vuelca en pantalla para evitar parpadeos, y *buffers* diversos para texturas y operaciones complementarias de todo tipo. De ahí que esta cifra inicial deba multiplicarse por varias unidades hasta alcanzar los 2, 3 o 4 Gb, que son relativamente corrientes en



las tarjetas actuales. Por las mismas razones que antes, tampoco es demasiado significativa la velocidad del reloj interno de la memoria que suele estar en torno a los 1.800 MHz. Pues es más significativo el tipo de memoria, la anchura de datos y la anchura de banda. El tipo de memoria de las tarjetas actuales suele ser DDR3 o DDR5. Esta denominación se refiere a un tipo de memoria que apareció hacia 2003 y cuyo nombre completo era DDR SDRAM (*double data rate synchronous dynamic random access memory*) que permitía aumentar los tiempos de transferencia hasta duplicar aproximadamente la anchura de banda de la memoria SDRAM, que a su vez era una mejora con respecto a la vieja memoria RAM (*random access memory*), la memoria principal de la placa base en donde se grababan de modo aleatorio (de ahí el nombre, *random*) los datos que se necesitaban mientras se ejecutaba una aplicación del tipo que fuera. La primera memoria DDR alcanzaba 128 bits por pulso (64 bits de bus x 2) lo que, para una frecuencia de 100 MHz supondría 1.600 MB/s ( $12.800/8$ ). Después aparecieron las DDR2, DDR3, DDR4 y DDR5, que proporcionaban mayores tasas de transferencia. Resumiendo: las tarjetas actuales tienen interfaces de memoria del orden de los 128 bits (gama baja), 256 bits (gama media) y 384 bits (gama alta), lo que se corresponde con tasas de transferencia (que varían con las frecuencias del reloj de memoria) del orden de los 30, 150 o 280 MB/s. Obviamente, cuanto mayor sea la tasa de transferencia más eficaz será la memoria.

c) Un tercer factor importante es la *tecnología soportada*. El primer dato es la versión de DirectX y OpenGL que incluye una determinada tarjeta. Pero este dato es poco significativo porque lo habitual es que incorporen la última versión que, en el momento de escribir este texto, es la 11 de DirectX y la 4 de OpenGL. Pero en los últimos años han ido apareciendo otras aplicaciones dirigidas específicamente a sacar mayor partido del paralelismo. En concreto, en las tarjetas de Nvidia se pueden encontrar módulos de OpenCL y diversos

núcleos CUDA. Y en las AMD, la tecnología Stream Processor. En la sección siguiente volveré sobre este tema.

d) Por último, hay otros factores que también hay que tener en cuenta. La *conexión* o *bus de interface* con la placa base es un tema fundamental pero en el que hay pocas diferencias pues la mayoría de las tarjetas actuales incorporan la última o penúltima versión de un bus de conexión, PCI (*peripheral components interconnect express*) introducido en 2003 y que ha desplazado a los anteriores por su mayor tasa de transferencia (aunque algunas tarjetas actuales todavía usan AGP). Este bus tiene conexiones escalables (x1, x2, x4, x16) que duplican las tasas de transferencia. Las versiones actuales corrientes son PCIe 2.0 (desde 2007), con tasas de transferencia de 5,0 GT/s y PCIe 3,0 (desde 2010), con tasas de 8,0 GT/s. Las *conexiones* con otros dispositivos (comenzando por el monitor) se llevan a cabo mediante conectores externos que, en las tarjetas actuales son al menos de tres tipos: VGA (la más corriente, ya citada), HDMI (*high definition multimedia interface*) y DVI (*digital visual interface*). La *temperatura* que puede llegar a alcanzar la placa es otro factor a tener en cuenta, pues en los modelos más potentes se requiere una entrada de alimentación independiente para la tarjeta y un buen sistema de refrigeración. Y, en fin, el *coste*, es un factor fácil de entender. De un modo muy simplificado puede decirse que una tarjeta de gama baja costará, en 2014, algo menos de 100 €, una de gama media puede estar en torno a los 200 o 300, y una de gama alta puede superar los 600 es decir, tanto como un ordenador sencillo.

### Direct3D y OpenGL. Shaders. Otras tecnologías recientes

Como ya he avanzado en el apartado anterior, durante la década de 1990 aparecieron dos grandes sistemas de comunicación entre las tarjetas gráficas y las placas centrales que, en pocos años, se convertirían en imprescindibles. Estos sistemas se denominan



genéricamente API (*application programming interface*) y pueden definirse como un programa que proporciona un conjunto de recursos, funciones y procedimientos, que se utilizan como intermediarios para comunicarse con otro programa o, más exactamente, con programas muy específicos ligados a un dispositivo, la placa gráfica y con programas gráficos de propósito general.

Hasta principios de 2000 el líder era OpenGL, que estaba concebido como una aplicación muy genérica, independiente de cualquier sistema operativo, basada en un concepto novedoso en la época, el de una máquina de estados finitos (*finite state machine*), un sistema basado en una serie finita de estados, transiciones entre estados y acciones, adaptado a todo tipo de aplicaciones y muy elástico.

Sin embargo, a partir de 2002 y la aparición de la versión 9 de DirectX (con su módulo Direct3D concebido expresamente para aplicaciones de simulación 3D), basado en programación orientada a objetos que ya se había impuesto como un paradigma mucho más potente y más fácil de compartir, menos flexible pero ligado a un sistema operativo, Windows (que se imponería en todo el mundo favoreciendo así el uso de productos ligados a su sistema y, también, con un trabajo intenso en el perfeccionamiento del sistema) la situación comenzó a invertirse. Las mejoras introducidas por Microsoft, así como las alianzas estratégicas que llevó a cabo con algunas empresas, le permitieron mantenerse en una posición destacada hasta las últimas versiones. OpenGL ha tenido que hacer esfuerzos importantes para mantener el ritmo.

En cualquier caso, lo que importa es que ambas API han ido incorporando progresivamente todo tipo de funciones. Funciones tales como transformaciones proyectivas básicas, recortes (*clipping*), rasterización, controles diversos para almacenar y gestionar texturas, funciones de procesamiento de imágenes que se incorporan a métodos de *antialiasing* por hardware, recursos para subdividir y gestionar texturas que se ven sometidas a

procesos de magnificación (aumento de su tamaño propio) o minificación (reducción de su tamaño propio) en vistas en perspectiva, filtros diversos que disimulan los errores corrientes que se producen a lo largo de estos procesos, control automático de oclusiones, efectos atmosféricos tales como efectos de niebla, efectos de desenfoque en función de la profundidad, teselación geométrica virtual para crear efectos de relieve y desplazamiento, efectos de fusión (*blending*), y un largo etcétera.

Otro concepto fundamental para comprender cómo funcionan los sistemas actuales es el de *shader*. Un *shader* es un procedimiento algorítmico invocado por un programa para computar un valor o un conjunto de valores requeridos durante el proceso de representación. La potencia de este concepto es que puede especializarse en un determinado efecto (una propiedad material, como puede ser un tipo de reflejo, una aplicación de una textura, una propiedad del entorno tal como un efecto atmosférico, etc.) y combinarse con otros *shaders* para crear un efecto complejo.

Este concepto fue introducido por Robert Cook en 1984 y fue desarrollado extensamente en la compañía creada por Georges Lucas, que posteriormente se integraría en Pixar, en relación principalmente con el lenguaje RenderMan, para dar lugar a toda una serie de *shaders* que se clasificaron en diferentes categorías: de superficies, de luces, de efectos atmosféricos, etc. Cook organizaba sus *shaders* en *shader trees*, estructuras jerárquicas que admitían todo tipo de entradas (otros *shaders*) para combinarse en la cúspide en un resultado unitario, en colores asignados a grupos de píxeles.

El concepto se integró en la programación de las tarjetas gráficas dando lugar a, entre otros, dos módulos fundamentales: los *vertex shaders* y los *pixels shaders*.

Un *vertex shader* es un módulo que actúa sobre los datos iniciales de la escena, los vértices de los modelos geométricos de los objetos. En estos vértices se integra la información



principal: la posición geométrica, la orientación de las normales, la topología dada por su particular conexión con otros vértices, las coordenadas de textura y la asignación superficial del color material y otras propiedades básicas. Cada vértice es sometido, en función de las características de la escena y de la posición de la cámara virtual, a una serie de transformaciones geométricas, que acaban con su normalización y proyección sobre el módulo de rasterización previo a su proyección como imagen de salida y previo, también, a su modificación adicional por otros *shaders*. En las tarjetas modernas, todas estas operaciones se llevan a cabo en paralelo por módulos especializados que pasan sus valores a registros diversos, lo que incrementa exponencialmente la velocidad de procesamiento.

Un *pixel shader* (OpenGL utiliza el término *fragment shader* que, en cierto sentido, es más exacto, aunque utilizaré el otro, más corriente) es un módulo que recoge la información proporcionada por los *vertex shaders*, la combina con la información proporcionada por otros módulos, incluyendo la correspondiente a los cálculos de iluminación, y la convierte en datos de salida: en el color que se asignará a cada píxel de la imagen. Una manera más sencilla de describir este proceso es decir que recibe datos sobre el color que correspondería inicialmente a ciertos objetos, y los convierte en datos sobre el color que hay que asignar a esos objetos tras recibir la influencia del conjunto de la escena.

La programación de estos y otros *shaders* exigía a los programadores utilizar los lenguajes incorporados a las API, principales, lo que incrementaba el tiempo dedicado a trabajos rutinarios de especificación de funciones que tenían poco que ver con lo fundamental. Como resultado de esta situación han ido apareciendo lenguajes de *shaders*, *shading languages*. El primero y más influyente fue RenderMan, que apareció en 1988. Pero este lenguaje está ligado a aplicaciones muy específicas propias de la empresa que lo desarrolló (Pixar).

Así que, como sería previsible, cada una de las dos grandes API ha desarrollado su propio lenguaje de *shaders*. OpenGL introdujo el GLSL (*openGL shading language*) cuya primera versión definitiva apareció en 2002, como extensión de la versión 1.4, aunque su aparición oficial se dio a partir de la versión 2.0, para proporcionar apoyo a los programadores interesados en crear módulos para OpenGL. Y Microsoft introdujo el HLSL (*high level shading language*) el mismo año, 2002, junto con Direct3D 9.0. Ambos lenguajes tienen características muy similares y se basan extensamente en la programación de *vertex shaders* y *pixel shaders*.

Por añadidura, hacia 2005 aproximadamente, comenzó a verse con mayor claridad que las dos grandes API tenían limitaciones importantes en su estructura que no permitía sacar todo su partido de la estructura de las tarjetas gráficas con su gran potencial para el procesamiento en paralelo. Y comenzaron a incorporarse técnicas que se habían desarrollado en otros campos, a finales del siglo xx, y que se denominaban como propias de procesadores de flujos (*stream processors*). Así surgió hacia 2000 lo que se denominaría GPGPU (*general purpose computing on GPUs*). Las dos grandes compañías que desarrollaban tarjetas gráficas, NVIDIA y AMD, desarrollaron, casi al mismo tiempo, módulos especiales que se adaptaban a este paradigma. En 2006, AMD incluyó con sus nuevas series, el módulo AMD *Stream Processor*, que se utilizó en campos científicos y financieros, entre otros. Y en 2007, NVIDIA lanzó la primera versión de CUDA (*computer unified device architecture*) un conjunto de herramientas que permitían desarrollar algoritmos específicos para iniciar varios hilos (*threads*) de ejecución en paralelo. Estas herramientas estaban diseñadas de tal modo que los programadores de *shaders* podían despreocuparse de la ejecución efectiva de sus algoritmos dejando en manos de estos módulos su descomposición en procesos paralelos que aumentaban de modo espectacular la eficacia y velocidad del proceso.





Por último, en 2009 apareció OpenCL (*open computing language*), un lenguaje de computación abierto, disponible para ambas plataformas y que también estaba especialmente preparado para explotar el paralelismo y acceder a recursos heterogéneos.

### Las tuberías de la representación

Todo lo anterior se aplica a un proceso cuyos estadios se han ido estandarizando progresivamente en los últimos 20 o 30 años. Este proceso discurre a lo largo de lo que se denomina corrientemente “tuberías de la representación”, una traducción bastante literal del inglés *rendering pipeline*. Otro modo muy sencillo de describir este proceso es decir que es el conjunto de operaciones normalizadas que se deben llevar a cabo para pasar de una estructura virtual 3D, que describe entidades geométricas asociadas a definiciones de propiedades materiales y de luces virtuales, a una imagen sintética 2D.

Como en los casos anteriores describiré muy sintéticamente estos estadios. Puede encontrarse una descripción más amplia, incluyendo las peculiaridades con que se aplica en Direct3D y OpenGL en el libro sobre simulación visual de materiales.

Las operaciones se describen en secuencia para clarificar la exposición. Pero, tal como se ha indicado en el apartado anterior, en las tarjetas gráficas modernas muchas de ellas discurren en paralelo.

**0 Procesamiento y evaluación de la información de entrada.** Los datos recibidos por el programa principal se someten a una serie de procesos que definen el contexto (la tarjeta concreta y otros detalles mecánicos y lógicos), y que concluyen con la recopilación normalizada de toda la información que se pasará al primer estadio.

**1 Transformación de vértices.** El dato inicial principal es la posición geométrica de cada vértice, su relación con otros vértices y sus atributos asociados o derivados, tales como el color asignado por la aplicación superior, la normal derivada de la agrupación

de vértices para formar un plano o las coordenadas de textura que se hayan definido para este vértice, si es el caso. Estos datos vienen definidos en el espacio objeto (*object space*) con un origen de coordenadas 0,0,0, que puede ser el centroide del objeto o un punto singular de este objeto. Las unidades concretas acabarán siendo normalizadas, reducidas a un rango de 0,0 a 1,0. Además de las coordenadas del espacio objeto se necesita un espacio común que se denomina espacio global (*world space*) y cuenta con su propio sistema de coordenadas, el sistema de coordenadas global (*world coordinate system*), con un origen que se especifica por el programa superior. Las coordenadas de los objetos se transforman en las coordenadas del espacio global mediante una transformación específica denominada transformación de modelado (*modeling transformation*) que implica una serie de transformaciones concatenadas.

**2 Cálculo básico de la iluminación.** La información suministrada por la aplicación incluye corrientemente la posición, intensidad y color de luces. Estas luces modifican el color de los vértices a partir de un cálculo elemental. Véase el apartado sobre este tema más adelante.

**3 Ensamblaje de primitivas.** Los datos iniciales asociados a los vértices se agrupan en primitivas: triángulos en la mayoría de los casos, que se recortan o se eliminan a partir de operaciones adicionales de reordenación, que incluyen sistemas sofisticados de indexación para evitar redundancias y que se envían a un registro especial, el *index buffer*.

**4 Transformación perspectiva. Culling.** Concluida esta reordenación previa, el espacio modelo se proyecta sobre un espacio perspectivo con un nuevo origen, que corresponde al punto de vista definido desde el programa. Las nuevas coordenadas se denominan coordenadas de ojo o de cámara (*eye space* o *camera space* y *eye coordinates* o *camera coordinates*). Por añadidura, las coordenadas de todos los objetos se transforman en coordenadas homogéneas y se

reproyectan sobre un espacio normalizado, un cubo unitario, con lo que todos los objetos tendrán coordenadas comprendidas entre (-1,-1,-1) y (1,1,1). Este espacio se denomina *NDS*, *normalized device space* y sus coordenadas correspondientes *NDC*, *normalized device coordinates*. En este estadio se lleva a cabo también un proceso denominado *culling* o *backface culling* en que se descartan (*culling*) los polígonos cuyas normales forman un ángulo mayor de 90° con el vector correspondiente al punto de vista, dando por supuesto que corresponderán, en general, a caras no visibles de un volumen cerrado.

**5 Clipping (*frustum clipping*).** El espacio resultante queda limitado por cuatro planos correspondientes a la pirámide de visión. Y todos los objetos o partes de objetos que queden fuera del espacio dado por esta pirámide se eliminan. Además, si el usuario ha especificado planos de recorte adicionales, perpendiculares a la línea de visión (*clipping planes*), los objetos situados fuera de estos planos también se eliminan. El espacio más general tiene, por tanto, 6 planos, cada uno de los cuales define un hemiespacio a un lado del cual se eliminan todos los objetos. Este espacio se denomina *frustum* (del latín fragmento, trozo).

**6 Transformación proyectiva 2D (*viewport mapping*).** Los objetos que caen dentro del *frustum* se proyectan sobre la ventana de salida cuyos píxeles también se normalizan y quedan definidos en el espacio de la ventana (*window space*) con sus correspondientes coordenadas de ventana (*window coordinates*). Sus valores están comprendidos entre (0,0) y (x, y), siendo x e y las dimensiones de la ventana. El origen varía según las API. En OpenGL está en la parte inferior izquierda. En Direct3D está en la parte superior izquierda de la pantalla.

Esta última transformación, que se lleva a cabo por una matriz específica, la matriz de visión se denomina proyección o transformación de visor o *viewport mapping* o *viewport transformation*.

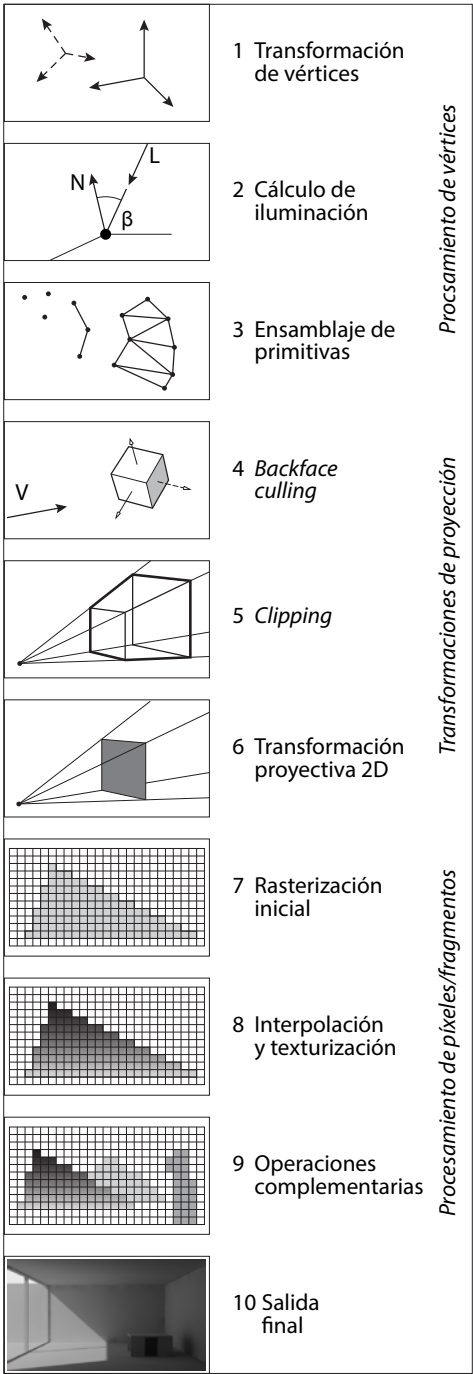


Figura 4.1 Rendering pipeline.



**7 Rasterización inicial.** Como resultado de todo los pasos anteriores, a cada primitiva le corresponderá un conjunto de píxeles o fragmentos. Direct3D utiliza el primer término y OpenGL el segundo, que puede considerarse más exacto pues no siempre este resultado previo, potencial, se corresponde con un píxel de salida final. Los píxeles/fragmentos pasarán todavía por una serie de operaciones antes de ser enviados al *buffer*.

### 8 Interpolación y texturización.

Los valores resultantes no abarcan todos los píxeles (o fragmentos) sino solo un subconjunto de la totalidad de la imagen. Los no computados directamente se interpolan a partir de los computados. Por añadidura, en este estadio también se completa la proyección de texturas, uno de los aspectos en que más se ha avanzado pues la gestión de texturas consume muchos recursos y, para gestionarlos desde la tarjeta gráfica, se han desarrollado funciones cada vez más complejas que permiten llevar a cabo todo tipo de substituciones, recortes, fusiones, combinaciones diversas, transformaciones de traslación, rotación, inversión o cambio de escala, por medio de matrices específicas, etc. Los valores resultantes de la aplicación de estas funciones se combinan por añadidura con otras entradas debidas al cálculo de iluminación avanzada o a otras modificaciones. El resultado de este proceso es que cada píxel recibirá una reasignación de color.

**9 Operaciones complementarias de rasterización.** En el caso de que el programa principal incluya especificaciones adicionales que puedan estar incluidas en la tarjeta, se activan en este último estadio. Algunas de las principales son las siguientes: *depth test* (los píxeles cuya coordenada *z* es mayor que la de otro píxel en la misma posición se eliminan). Este test corre a cargo de un sector de memoria especial, el *z-buffer*, que almacena estos valores para acelerar este procesamiento o, en las tarjetas modernas, otro *buffer* más preciso denominado *w-buffer*; *stencil test* (ciertos píxeles se enmascaran en fun-

ción del contenido del *stencil buffer*, que actúa como una plantilla superpuesta al área de *rendering*); *blending* (hay píxeles que pasan las pruebas anteriores de un modo parcial, lo que quiere decir que deben combinarse con valores que provienen de otro canal por medio de operaciones de adición, sustracción o multiplicación, entre otras). Además de estas, que son las más corrientes, hay otras operaciones. Por ejemplo: *pixel ownership test*, *scissor test*, *occlusion query*, *sRGB conversion*, entre otras.

**10 Salida final.** Los resultados que sobreviven a todas estas pruebas se envían al *back buffer* que mantiene la información preparada para ser enviada al *front buffer*, a la salida final. Cabe todavía la posibilidad de fusiones adicionales por medio de operaciones que se denominan, en Direct3D, *render target blending*.

En cualquier caso, el resultado de todo este proceso es una imagen que se envía a la pantalla del dispositivo concreto que se vaya a utilizar.

## Los cálculos de iluminación y la pipeline

### Cálculos estandarizados

Con independencia de los cálculos globales de iluminación avanzada, que dependen del programa principal, para trabajar adecuadamente con un modelo virtual se requiere una interacción fluida con el usuario. Esto quiere decir que la escena debe quedar iluminada de un modo que responda suficientemente a las características de la iluminación diseñada y que los cálculos correspondientes deben realizarse en tiempo real. Este cálculo, que debe realizarse a una velocidad muy superior a la de la representación final de la escena, corre a cargo de la GPU.

En los casos más elementales, por ejemplo, al arrancar el programa y visualizar un modelo que se acaba de crear o importar, todo se basa en tres mecanismos muy simples. En primer lugar, hay que activar o desactivar la



iluminación. El resultado aparentemente paradójico es que, si se activa la iluminación pero no hay luces en la escena, todo aparecerá negro. Pero si se desactiva, sin luces en la escena, aparecerá iluminado. Pues, en principio, se utiliza un sistema de luces predeterminado, al igual que ocurre en muchos programas comerciales de simulación.

Suponiendo, por tanto, que queramos utilizar nuestras propias luces, el primer paso es activar la iluminación creando alguna luz. Una vez que la iluminación está activada, se pueden utilizar dos modos simples de iluminación: *ambient lighting* y *direct lighting*. Con *ambient lighting*, un método que ya está en desuso en programas avanzados de simulación pero se mantiene por razones de compatibilidad y se utiliza en algún caso en video juegos, todo queda iluminado por igual lo que evita que aparezcan superficies completamente negras. Con *direct lighting* todo queda iluminado según una dirección dada por un vector. Cada uno de estos dos modos puede activarse mediante funciones simples semejantes a la anterior.

Por añadidura pueden crearse (o utilizarse) tres tipos de luces: luces puntuales (*point lights*), focales (*spot lights*) y direccionales (*directional lights*). Las luces puntuales tienen una posición dada en el espacio e irradian por igual en todas direcciones. Las luces focales tienen una posición dada e irradian en una dirección principal. Además, tienen un cono interno y un cono externo que determinan como se distribuye su iluminación. En la zona que abarca el cono interno la intensidad es máxima. En la zona comprendida entre el cono interno y externo la intensidad pasa de ser máxima a ser nula. Las luces direccionales no requieren una posición en el espacio e irradian en una dirección única. Véase la figura 2.4, correspondientes a luces simples, del capítulo 2 que se corresponde exactamente con estas luces estandarizadas incorporadas a la GPU.

Estas luces básicas, presupuestas por las aplicaciones que controlan las tarjetas gráficas, están ligadas a dos métodos básicos

de cálculo de la iluminación: *flat shading* y *smooth shading* (o *Gouraud shading*) utilizados tanto por Direct3D como por OpenGL y q.

Como también hemos visto, en el primer caso se computa el producto vectorial punto del vector Luz normalizado con el producto vectorial punto del vector Normal a la superficie igualmente normalizado. Como este producto es el coseno del ángulo formado por los dos vectores, y este coseno es 0 cuando el ángulo es  $90^\circ$  y 1 cuando el ángulo es  $0^\circ$ , resulta sencillo ajustar la intensidad en función de un valor lineal comprendido entre 0 y 1, lo que se corresponde exactamente con la menor o mayor incidencia de la luz sobre la superficie.

En el segundo caso se aplica el mismo procedimiento pero se envía la información adecuada al *pixel shader* para que interpole estos valores entre los vértices correspondientes. El resultado es que la superficie mostrará un degradado continuo en lugar de facetas planas, lo que producirá el efecto de que la superficie tiene una curvatura continua (un efecto traicionado por los contornos). Véase también las figuras 2.2 y 2.3, correspondientes a estos modos básicos en el capítulo 2.

Cada uno de estos modos depende por añadidura de las propiedades de los materiales (color difuso, ambiente, emisivo y especular) que modifican los valores obtenidos con el cálculo anterior. El cálculo de iluminación da lugar, por consiguiente, a que cada vértice cuente con un color primario (correspondiente a la definición de material) y un color secundario (correspondiente, en principio, a su modificación por la luz).

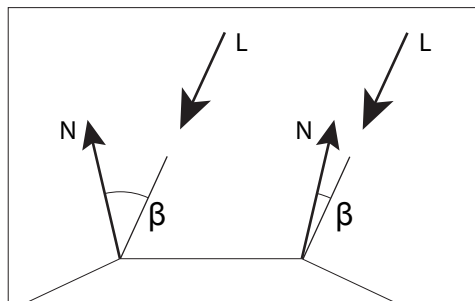


Figura 4.2 Cálculos básicos de iluminación.



## Iluminación global con la GPU

La descripción del apartado anterior es muy antigua y sigue siendo válida para una interacción elemental entre el usuario y el modelo. Pero, en principio, los sistemas avanzados no se pueden integrar en la GPU y el cálculo se realiza a través de la CPU. Esto hace que se produzca un salto considerable en el rendimiento. Resulta un tanto desconcertante estar interactuando con una escena en tiempo real y a toda pantalla y tener que esperar unos cuantos segundos cuando se presiona el botón de *render* para que el programa muestre un resultado que tampoco es tan distinto, en algunos casos, de aquel con el que estamos interactuando. Naturalmente, en otros casos, lo que vemos tiene una calidad mucho mayor y puede justificar la espera. Pero el retardo se produce, aunque sea mucho menor, también en casos más sencillos. Y esto es debido a que el cálculo pasa de la GPU, la tarjeta gráfica, a la CPU, la placa base del ordenador.

En los últimos años han surgido un gran número de trabajos que buscan integrar los métodos avanzados de cálculo, propios de lo que se denomina Iluminación Global o GI por sus siglas en inglés, en la GPU. Estos métodos también se denominan de GI “en tiempo real”. ¿Qué se entiende por “tiempo real”? En principio, esta frase alude a una respuesta “inmediata” del sistema, es decir, que cualquier cambio en la configuración de las luces (intensidad, posición, parámetros complementarios, etc.) se traduzca en un cambio “inmediato” de la imagen que representa la escena. Pero esto es una exigencia que puede rebajarse y algunos investigadores consideran que se puede hablar de “tiempo real” si la respuesta es inferior a 1 segundo. Dejémoslo aquí.

En cualquier caso, esto requiere reestructurar los algoritmos de estos sistemas para integrarlos adecuadamente en los recursos disponibles en las tarjetas gráficas más avanzadas, lo que no siempre es posible. Entre otras cosas porque la computación de la ilu-

minación indirecta requiere contar con información sobre la visibilidad, no solo desde la cámara sino también desde cada punto que contribuye al cálculo. Y las GPU no cuentan, por ahora, con semejante posibilidad.

Estos métodos están evolucionando con considerable rapidez y es probable que para cuando este se publique hayan surgido nuevos métodos. Lo que sigue debe verse, por tanto, como una introducción muy general a algunos de los principales. La presentación, tras el primer apartado más general, se hace aproximadamente por orden cronológico, a partir de la fecha del primer artículo principal citado.

### Aspectos generales. Ventajas y limitaciones del uso de la GPU. *Rasterization* frente a *Ray tracing*

En apartados anteriores hemos visto un esquema de lo que se denomina *rendering pipeline*. La arquitectura de las GPU en los últimos años se ha desarrollado a partir de este esquema, a partir de lo que se denomina corrientemente *rasterization*. Esto quiere decir que están optimizadas para este proceso, con diversos mecanismos, *buffers* de memoria, *shaders* integrados en hardware, etc., adaptados a este proceso. De este modo, se saca partido de las posibilidades de procesamiento en paralelo de la GPU que proporcionan rendimientos mucho más altos que los de la CPU, adaptando el proceso a esta arquitectura.

En realidad, cuando se utiliza *rasterization*, el tiempo de cálculo es lineal pues cada uno de los polígonos debe ser procesado. Sin embargo, el cálculo, pese a esto, es extraordinariamente rápido, con velocidades del orden de más de 500 millones de polígonos por segundo, en tarjetas gráficas relativamente corrientes. Esto se debe a que hay partes de la GPU que no hacen otra cosa que procesar polígonos. Y hay otras partes de la GPU, como los registros de *depth culling* o *hierarchical culling*, especializadas en descartar los polígonos no visibles desde la cámara, lo que aún acelera más el proceso.



Ahora bien, con la evolución de las GPU, estos sectores especializados han ido ocupando cada vez menos, en torno a un 5 % de la zonas principales de las GPU recientes. Y ha ido creciendo el número de registros que se pueden dedicar a otras cosas, tales como almacenar texturas, pero que también pueden almacenar otro tipo de datos si interesa hacerlo así, pues las texturas se registran como colecciones estructuradas de datos abstractos.

Y esto abre el camino a otras posibilidades. Pues la arquitectura especializada de las GPU no se adapta a los métodos avanzados de cálculo de iluminación. La diferencia principal está en que estos métodos procesan relaciones: el modo en que, por ejemplo, el color de un triángulo depende del color de otro triángulo cercano. Y el proceso de *rasterization* está especializado en un único tipo de relación, la que se establece entre vértices y píxeles. Pero hay otros aspectos que también dificultan esta adaptación y que están en vías de solucionarse. Entre los aspectos que se han solucionado o están en vías de solución, se pueden citar la falta de poder de computación, concretamente de operaciones en coma flotante, la falta de un conjunto adecuado de instrucciones o, en general, la falta de programabilidad. Estos aspectos se han resuelto o se han resuelto parcialmente, si bien siguen pesando, debido principalmente a que no hay acceso aleatorio a la memoria, lo que impide la construcción de la mayoría de estructuras de datos requeridas y, en consecuencia, dificulta la creación de algoritmos de búsqueda adecuados.

Para esquivar el problema del cómputo de relaciones sacando partido de la potencia de la GPU, muchos sistemas procesan diferentes aspectos por separado y luego los combinan. Una técnica muy extendida, y que puede servir para intuir el modo en que se usan otras técnicas que veremos más adelante, es lo que se denomina técnicamente *deferred shading*, es decir, un cálculo del color de las superficies diferido hasta un segundo pase. En el primer pase, las posiciones, normales y

propiedades de materiales de cada superficie se representan y se almacenan en un *buffer* especial denominado *G-buffer* (con g de geometría) como una serie de texturas, de modo muy similar a lo que se obtiene cuando se lleva a cabo lo que se denomina representación por elementos (*render elements*) un proceso incorporado a muchos programas de simulación para representar aspectos parciales de una escena (reflejos, sombras, iluminación, etc.) y componerlos más adelante con programas especiales. En el segundo pase, el *pixel shader* computa la luz directa e indirecta y lo combina con estas “texturas”. Este proceso desconecta el cálculo de la geometría del cálculo de iluminación y simplifica este último. El uso de diferentes *buffers*, por ejemplo, *buffers* de acumulación (*accumulation buffers*), permite sacar partido de la estructura de las GPU y evitar la codificación y procesamiento condicional de estructuras de datos complejas, algo que no es posible en la GPU.

Pero este tipo de recursos queda muy lejos de lo que se hace realmente en un sistema de iluminación avanzada. Y es importante comprender que las diferencias de planteamiento son fundamentales.

En el contexto de *rasterization*, el *shading* precalculado, sea cual fuere, está ligado directamente a la geometría de la escena. Como mucho, se puede elegir entre guardar los datos como vértices o como mapas, capturas de la escena desde una posición. Guardarlos como vértices tiene el inconveniente de que se hace depender la interpolación, en su caso, de la mayor o menor resolución de la malla. Guardarlo como mapas (*shadow maps*, *light maps*) tiene el inconveniente de que depende de la resolución del mapa y puede dar lugar a *aliasing*. Pero en ningún caso es posible un cálculo que compute de modo efectivo cómo recorre la luz todos los objetos incluidos en la escena y que posibilite una estimación de cómo se distribuye la iluminación a partir de un análisis de este recorrido. El método de representación es muy rápido debido a la especialización de las tarjetas. Pero es intrínsecamente ajeno a la precisión





y a las posibilidades de un sistema de cálculo avanzado

En el contexto de *ray tracing* se pueden utilizar métodos que sean independientes de la geometría, de las mallas poligonales y sus vértices y envíen rayos hasta las zonas más recónditas para captar la influencia de la luz en estas zonas. En la mayoría de los sistemas de GI, el uso de rayos trazadores, de un tipo u otro, permite almacenar puntos de superficie según lo que interese al sistema, con independencia de si la superficie es más o menos compleja, si está compuesta por más o menos triángulos. Con *photon mapping*, por ejemplo, se almacenan mapas de fotones que dependen del número de fotones que se quiera utilizar, no del número mayor o menor de vértices geométricos en la escena concreta que se está procesando. Por otro lado, el uso de *ray tracing* para encontrar puntos de intersección con las superficies requiere calcular estas intersecciones. En principio, *ray tracing* tendría también la ventaja de que puede calcular la intersección entre un rayo y una primitiva dada por una fórmula general. Por ejemplo, una esfera puede venir dada por una ecuación simple, lo que ahorraría tener que computar la intersección con las docenas o centenares de pequeños triángulos con los que se la representan habitualmente. Pero en la práctica esta ventaja es pequeña, sea porque las primitivas vienen ya dadas como series de triángulos, sea porque en muchas escenas complejas las superficies no pueden representarse por primitivas. Y también resulta más sencillo utilizar un sistema de *ray tracing* si el algoritmo solo contempla triángulos como superficie de intersección. En cualquier caso, esto obliga a los métodos de *ray tracing* a utilizar estructuras secundarias que envuelven a los objetos para llevar a cabo un proceso previo, que descarte zonas de la escena y ahorre cálculos innecesarios. Como ya hemos visto en el capítulo 3, buena parte de la literatura sobre *ray tracing* está dedicada a construir y optimizar este tipo de estructuras. Para resumir: los métodos de *ray tracing* son más exactos y se adaptan mejor a su

integración en algoritmos eficaces pues son más flexibles y pueden aplicarse de modos muy diferentes. También permiten calcular con mayor precisión todo tipo de efectos, tanto de representación de materiales como de procesamiento de elementos geométricos, como de iluminación. Pero no se adaptan fácilmente a las GPU actuales.

La pregunta obvia es: si esto es así, es decir, si los métodos de *ray tracing* tienen, en general, más ventajas que los métodos de rasterization ¿por qué las tarjetas gráficas no se han construido de modo diferente, para adaptarlas a *ray tracing*? La respuesta, ya indicada al comienzo de este apartado, es que las tarjetas gráficas comenzaron su acelerada evolución antes de que aparecieran estos métodos y que, incluso cuando aparecieron, la aplicación de estos métodos era muy especializada y quedaba, entonces, muy lejos de lo que eran (y siguen siendo hasta cierto punto) las aplicaciones más corrientes.

La siguiente pregunta obvia es: dado el espectacular desarrollo de las tarjetas gráficas y la creciente demanda de aplicaciones más realistas ¿es previsible que la construcción de tarjetas gráficas evolucione para dar cabida a estos nuevos métodos? Y la respuesta, en este caso, es que ya está sucediendo y es previsible que siga ocurriendo, si bien el proceso puede ser lento debido a la gran inercia creada sobre la base de los métodos de *rasterization*. Así que, al menos por algunos años, parece previsible que, por un lado, se sigan desarrollando métodos ingeniosos para sacar partido de las GPU actuales modificando los algoritmos originales y, por otro lado, se desarrollen GPU de alto nivel que permitan incorporar con mayor facilidad los algoritmos clásicos sin grandes modificaciones, como también ya está ocurriendo.

Así, los métodos generales de *ray tracing* se aplican, en la actualidad, de tres modos: a) Por medio de la CPU, con aplicaciones optimizadas de los métodos generales que ya hemos visto. Esto tiene la ventaja de que ofrece la máxima flexibilidad pues la CPU no impone ninguna restricción sobre el tipo de programa





utilizado. Pero tiene la desventaja de que la mayoría de las CPU se concentran en la aceleración del hilos (*threads*) únicos y, por tanto, no puede explotar el paralelismo implícito en *ray tracing* que permitiría aumentar espectacularmente su velocidad de ejecución; b) Por medio de GPU programables, con aplicaciones de los métodos generales, adaptadas a los recursos de procesamiento en paralelo masivo de estas tarjetas. Este paralelismo de la arquitectura propia de la GPU es su mayor ventaja. Pero dicha arquitectura, que ha evolucionado para adaptarse a los procesos de *rasterization*, es, por esta razón, muy restrictiva en cuanto a su uso y debería modificarse radicalmente, lo que no es previsible que ocurra a corto plazo y, en cualquier caso, es improbable que llegue a ser tan flexible como la CPU. Debido a ello se requiere modificar la estructura algorítmica de *ray tracing* para adaptarla a la GPU; c) Por medio de GPU especializadas, diseñadas de modo expreso para adaptarse a los métodos generales de *ray tracing* con escasas modificaciones. Pero el inconveniente en este caso es el alto coste y escasa disponibilidad de estos recursos, algo que tampoco es probable que cambie demasiado en los próximos años.

En el apartado correspondiente a *ray tracing* describiré los principales métodos recientes del segundo grupo, es decir, métodos que adaptan los algoritmos generales de *ray tracing* a la estructura de las GPU más o menos corrientes. Insisto en que la intención de este y los demás apartados no es sino proporcionar una visión general del modo en que estas técnicas están evolucionando y es previsible que evolucionen en el futuro inmediato. No debe perderse de vista que ni la separación entre *rasterization* y *ray tracing* es tajante, pues muchos de estos métodos sacan partido de una hábil combinación de ambas técnicas, ni las técnicas de *ray tracing*, como también he insistido, deben verse como un método completo en sí mismo, algo que raras veces ocurre, sino como un método que se combina con múltiples variantes, con estrategias muy diversas.

### Múltiples luces (VPL). *Instant radiosity*. *Lightcuts*. Otras variantes

Lo que se denomina radiosity instantánea, *Instant radiosity*, IR, se presentó por primera vez en 1997, en un artículo de Alexander Keller (Keller, 1997). El término se presta a confusión pues tiene menos que ver con radiosity que con otras técnicas, como *photon mapping*. Pero esto es probablemente debido a que esta última acababa de aparecer y aún no era suficientemente conocida, a diferencia de radiosity que seguía siendo la referencia principal por aquellos años. En cualquier caso, el término ha quedado como una de las referencias principales para las técnicas que siguieron de procesamiento de iluminación avanzada en tiempo real, “instantáneo”.

Es una técnica de dos pases que concentra la energía radiante en determinados puntos de la superficie de los objetos de la escena, creando nuevas luces virtuales en esos puntos. De este modo, se evitan los principales inconvenientes de radiosity: hay una discretización explícita, no hay factores de forma complejos y se pueden utilizar luces puntuales simples. Por otro lado, los itinerarios, los *paths* que se forman de este modo, pueden reutilizarse. Y al utilizar luces simples, incorporadas a los métodos corrientes de la GPU, como ya hemos visto, se pueden acelerar por hardware. En principio, se asume que las superficies son lambertianas pero las reflexiones especulares se pueden gestionar por otros métodos complementarios.

El algoritmo se desarrolla en las siguientes fases que se ilustran esquemáticamente en la figura 4.3:

- a) En el primer pase se envían rayos desde la fuente de luz a la escena. Los vértices de los *paths* se tratan como luces puntuales virtuales, VPL (*virtual point lights*) En este estadio, se genera una aproximación, por medio de partículas, a la integración de la ecuación de *rendering*, considerando únicamente la radiancia difusa y utilizando QMC (*quasi Monte Carlo*) como método de cálculo, combinado con una variante de se-

cuencias de baja discrepancia clásicas. El número de luces utilizadas por Keller era pequeño, del orden de 50 a 300.

- b) En el segundo pase la escena se representa varias veces, desde cada fuente de luz (las originales y las virtuales). Esta re-

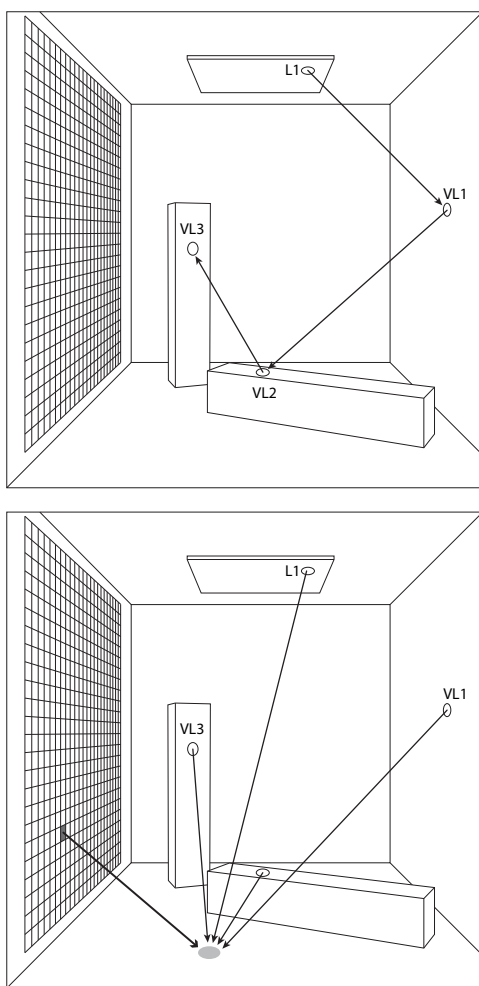


Figura 4.3 Instant radiosity: a) Primer pase en que se envían rayos desde las fuentes de luz a la escena. En los vértices de los paths se crean luces puntuales virtuales (VPL); b) Segundo pase en que la escena es representada varias veces por cada luz. La GPU procesa cada imagen, con sombras y reflejos, por cada partícula utilizada como fuente de luz. La imagen final se compone en un buffer especial (accumulation buffer).

presentación se desarrolla por hardware y computando las sombras para cada partícula utilizada como fuente de luz.

- c) La imagen resultante se compone en la GPU por medio de un *buffer* de acumulación (*accumulation buffer*), una zona de memoria en la que se guarda información sobre cada píxel antes de componerla en la imagen de salida final.

El algoritmo se complementa con métodos de *antialiasing* que eliminan (hasta cierto punto) los defectos del proceso. En la fase de supermuestreo se utilizan secuencias bidimensionales de Hammersley con desplazamientos aleatorios (*two-dimensional jittered Hammersley sequences*). En la fase del pase aleatorio (*quasi random walk*) se utilizan secuencias multidimensionales de Halton. Esto implica mayor control adaptativo aunque requiere comprobar si el proceso se degrada cuando las dimensiones son muy grandes debido a múltiples rebotes.

La principal ventaja de este método, comparado con radiosity clásica, es que no modifica la malla geométrica inicial lo que, entre otras cosas, evita la aparición de artefactos característicos debido a la discretización de la malla, que eran característicos de la radiosity clásica.

Los principales inconvenientes son que ya no es *unbiased* (no converge hacia resultados físicamente correctos) y que hay pérdidas de energía en las aristas lo que da lugar a manchas difíciles de controlar.

En 2005, Bruce, Walter, Sebastian Fernandez, Kavita, Bala y otros autores (Walter *et al.*, 2005) presentaron una variante de este método que denominaron *lightcuts*. A diferencia del método de Keller, permitía procesar cientos de miles de luces puntuales virtuales utilizando solo unos pocos cientos de rayos de sombra. Las luces se organizaban en un árbol para optimizar el procesamiento. Es decir, que se representaba la iluminación indirecta como una nube de puntos jerárquica. El algoritmo era escalable, lo que permitía manejar un gran número de luces de modo eficiente. La visibilidad se calculaba por *ray tracing*.



Una de las principales diferencias con el método de Keller es el uso de agrupaciones o *clusters* que permiten adaptar el resultado a lo que interese por medio de particiones o cortes, *cuts*, en la estructura de luces virtuales generadas previamente. El algoritmo general se basa en lo siguiente: dado un conjunto de puntos de luz  $S$ , su radiancia  $L$ , causada por su iluminación en un punto superficial  $x$ , visto desde la dirección  $w$ , es el producto de los términos que representan sus atributos materiales, su geometría, su visibilidad y su intensidad. Es decir:  $L_s(x, w) = f(M, G, V, I)$ , donde  $M(x, w)$  es la parte correspondiente a sus atributos materiales,  $G(x)$  es el factor correspondiente a la geometría del punto considerado,  $V(x)$  es la visibilidad del punto considerado e  $I$  su intensidad. El coste de este cómputo, que debe llevarse a cabo para cada luz, es, en principio, lineal. Para crear un método escalable, sublineal, se necesita agrupar las luces de modo que pueda aproximarse su contribución global sin necesidad de llevar a cabo el cálculo para cada luz. Para ello se define un *cluster* como un subconjunto de  $S$ , los puntos de luz, como un conjunto de luces que incorpora una luz representativa,  $j$ , que es parte del conjunto. La iluminación directa de un *cluster* puede aproximarse utilizando los términos  $M, G, V$  de esta luz para todos los miembros del *cluster*. La intensidad de todas las luces del *cluster* puede precomputarse y almacenarse con el *cluster*.

Para generar *clusters* que se adapten bien a la escena se utilizaba un *light tree*: un árbol binario en el que las hojas son luces puntuales y los nodos interiores son *light clusters* que contienen las luces que están bajo ellos. Un *cut* a través de los nodos es un conjunto de nodos tal que cada itinerario desde la raíz del árbol hasta una hoja (una luz) contendrá exactamente un nodo del *cut*.

La discusión se centra en el modo más eficaz de llevar a cabo esta subdivisión. Remitiéndose a la ley de Weber y al hecho experimental de que solo percibimos modificaciones de un 1 % en el contraste (como ya hemos visto en el capítulo 1), los autores parten del hecho, comprobado mediante pruebas de su

algoritmo, de que modificaciones de hasta un 2 % no dan lugar a errores perceptibles. El proceso comienza, por tanto, con un *cut* muy básico que se va refinando progresivamente hasta que el error alcanza el límite deseado. Se utilizan tres tipos de luces *omni*, *oriented* (*spots*) y *directional*. Para cada tipo se crea un árbol distinto aunque se considera que las tres forman parte de un árbol global. Para el término material se consideran también tres tipos de BRDF, Lambertian, Phong e Isotropic Ward.

El método puede extenderse a casos más complejos tales como luces de área, que se simulan por medio de múltiples luces, mapas de entorno HDR, que también se simulan con hasta 3000 luces puntuales, etc. A diferencia de *instant radiosity*, pueden utilizarse “miles o incluso millones” de luces virtuales.

La principal ventaja del método es su escalabilidad. Tal como se muestra en las pruebas que se resumen en la sección 6 del artículo original, al aumentar el número de luces los tiempos crecen muy suavemente en comparación con los métodos clásicos. El lector interesado puede consultar este apartado del artículo original para datos concretos sobre número de luces y tiempos.

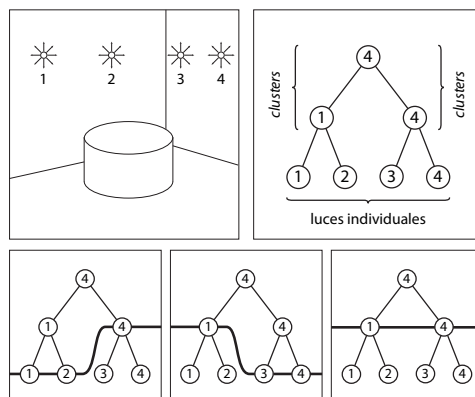


Figura 4.4 Lightcuts. Organización de las VPL: a) Vista de la escena con 4 luces puntuales virtuales (VPL); b) Organización de las luces en un árbol jerárquico y en clusters; c) Diferentes particiones (*cuts*) del árbol.



En (Segovia *et al.* (2006) se propone otra alternativa, *bidirectional instant radiosity* que tenía algunas ventajas importantes. Optimizaba la localización de las VLP, buscando posiciones que tuvieran influencia en la iluminación de la escena vista desde la cámara. Para ello, primero se enviaban rayos desde la cámara y, luego, los vértices de *paths* de longitud 2 formaban el conjunto de lo que se denomina *reverse VPL* (candidatos inversos para las VLP). Por último, se conectaban los *reverse VLP* con los VLP estándar. Tanto este método como el que sigue, permitían acelerar el cálculo al aumentar el número de VLP en áreas que contribuían mucho al resultado final y reducirlas en las que contribuirían menos.

Y al año siguiente, en (Segovia *et al.*, (2007) se proponía otra alternativa más, *metropolis instant radiosity* que se basaba en lo siguiente. Se buscaban VPL que iluminasen partes de la escena vistas desde la cámara. Para ello: a) primero se utilizaba la secuencia estándar de MLT (*metropolis light transport*) para muestrear VPLs. Esto es lo que se denominaba “la parte MLT”; b) luego, para cada *path*, se almacenaba el segundo punto como VPL; c) por último, se acumulaban las contribuciones de todas las VPL. Esto es lo que se denominaba “la parte IR” (*Instant Radiosity*).

En general, con cualquiera de estos métodos, la calidad del resultado depende del número de VPLs que se generen. En la práctica, el número de VLPs requeridos es un orden de magnitud menor que el de fotones que harían falta para un cálculo similar.

El mayor coste del procedimiento introducido por Keller es el que resulta de construir un *shadow map* para cada VPL por lo que gran parte de la investigación posterior fue dirigida a optimizar este proceso. Menciono para terminar otras dos contribuciones bastante citadas que pueden ilustrar las múltiples variantes a que ha dado lugar este método.

En (Hasan *et al.*, 2007) se utiliza una representación de las múltiples luces virtua-

les, VLP, por medio de una matriz compleja en la que cada columna representa todas las muestras iluminadas por luces individuales y cada fila una muestra individual iluminada por todas las luces. Una solución de fuerza bruta consistiría en acumular todas las columnas y, así, sumar la contribución de todas las luces para todos los píxeles. Pero esta matriz es, a menudo, de bajo rango lo que permite revelar su estructura por medio de un subconjunto pequeño de filas y columnas.

En Dachsbacher y Stamminger (2005) se presenta otro método que los autores denominaron RSM, *reflective shadow maps* y que saca partido de la capacidad de *radiosity* de la GPU para acelerar el proceso a costa de varias simplificaciones. Un *reflective shadow map* es una extensión de un *shadow map* en el que cada píxel se considera como una fuente de luz indirecta. La iluminación dada por estas luces se evalúa sobre la marcha (*on the fly*) por medio de muestreo adaptativo en un *pixel/fragment shader*. Utilizando interpolaciones en el espacio de la imagen de salida se consigue acelerar el proceso hasta conseguir que sea interactivo. Dado que el proceso se lleva a cabo principalmente en este espacio, es básicamente independiente de la complejidad de la escena. Los resultados son aproximados pero convincentes. La idea se basa en la observación de que, si tenemos una luz puntual, toda la iluminación indirecta dada por un solo rebote es debida a las superficies visibles desde el *shadow map*. En este caso, el *shadow map* contiene toda la información requerida acerca de la luz indirecta y no se necesita llevar a cabo otro tipo de procesos más complejos para calcularla. Se asume de entrada que todas las superficies son reflectores difusos y el mapa almacena, para cada píxel, además del valor de profundidad, la posición en coordenadas globales, la normal y el flujo radiante reflejado del punto visible de la superficie. El flujo define el brillo y la normal, las características de la emisión espacial.



### **Precomputed Radiance Transfer (PRT). Otros métodos de precómputo**

Los métodos de precómputo tienen una larga tradición en aplicaciones de simulación visual pues hay recursos, utilizados desde hace muchos años pero que siguen siendo habituales en juegos de vídeo, como los *light maps*, que consisten en grabar la iluminación como textura para sacar partido de la rapidez de procesamiento de las texturas de las GPU, lo que supone suprimir o reducir al mínimo el cálculo de iluminación. Pero el uso del término, en el contexto general de la aplicación de técnicas de GI a la GPU, se reserva para métodos menos artesanales y que permitan relegar el cálculo de iluminación en aplicaciones interactivas.

La primera referencia importante, y la más citada, es el artículo publicado por Peter Sloan, Jan Kautz y John Snyder en 2002 (Sloan *et al.*, 2002) donde se presenta un método para representar objetos que simulen materiales difusos y reflectantes con iluminación avanzada. La principal ventaja del método es que proporciona un recurso para aproximarse con gran rapidez a la iluminación global de una escena estática con luces complejas que pueden cambiar de posición. Se pueden integrar en el método luces de área, sombras suaves y luces indirectas complejas. También puede incorporar, mediante funciones adicionales, reflejos y sombras producidas por objetos sobre sí mismos. El principal inconveniente del método es que requiere un tiempo bastante largo de preprocesamiento, del orden de varios minutos o incluso horas en la época en que se propuso. Pero sus ventajas han hecho de él una referencia que sigue siendo importante, no solo desde el punto de vista práctico sino por el uso de técnicas de interés teórico.

La idea básica, común a otros métodos, es hacer un cálculo previo de una parte de la *rendering equation*, concretamente, la que afecta a las superficies difusas. O bien, dicho de otro modo, dividir la *rendering equation* en dos partes, una que se resuelve en un

pase precomputado y otra que se resuelve en tiempo real. Para ello se aceptan dos supuestos iniciales: a) ningún objeto de la escena emite luz, b) las fuentes de luz están en el infinito. Este segundo supuesto implica que la dirección de la luz incidente sea independiente de la posición del punto  $x$ . Por añadidura, se supone, en principio, que todas las superficies son difusores perfectos (lambertianos) lo que permite simplificar aún más la ecuación. Las funciones  $R$  (reflectividad),  $V$  (visibilidad) y  $G$  (el factor de forma geométrico o relación entre puntos cercanos) se agrupan en una función única,  $T$ , denominada *transfer function*, que es el producto de estas tres funciones.

Un aspecto crucial del método es que las funciones de transferencia se representan por medio de armónicos esféricos (SH, *spherical harmonics*).

Un modo simple de resumir lo que son las SH es decir que es un recurso que permite representar señales sobre una esfera. Y que lo hace, como en el caso general de las señales, por medio de series de Fourier, es decir, puede decirse que son el equivalente, sobre la superficie de una esfera, a lo que las series de Fourier son sobre una línea. Dicho de otro modo, es un método muy adecuado para representar valores que varían según su dirección, como ocurre precisamente con los valores lumínicos. Un modo algo menos simple de explicarlo es comenzar por recordar que se puede representar una función cualquiera como la suma de funciones de base ponderadas. En el caso de las SH, esto implica que hay una reparametrización de la base que lleva a expresar los valores por medio de funciones trigonométricas que, básicamente, dependen del ángulo de elevación ( $\theta$ ) y del ángulo azimutal ( $\varphi$ ) sobre la esfera. Esto tiene varias ventajas técnicas en la medida en que se adapta mejor a los tipos de proyecciones que interesan. Las funciones de base de SH son ortonormales, lo que simplifica los cálculos, tienen invariancia rotacional y permiten adaptarlas a frecuencias más bajas o más altas en función del número de términos utili-

zados, lo que, entre otras cosas, también permite manejar de modo adecuado el *aliasing*.

Al igual que ocurre con las series de Fourier, los armónicos esféricos son series infinitas por lo que es preciso cortar los órdenes o bandas utilizadas, teniendo en cuenta que se comienza por 0 y cada nueva banda  $B$  añade

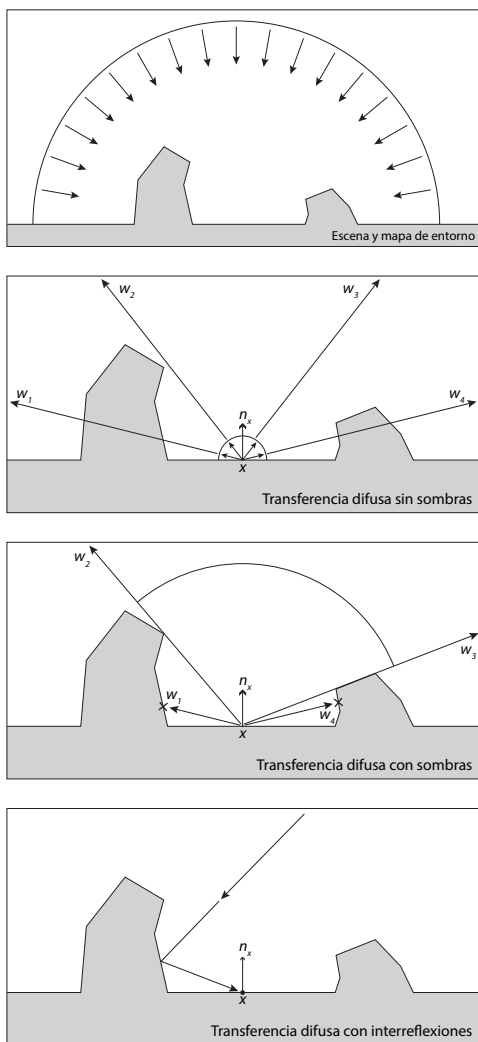


Figura 4.5 Precomputed radiance transfer. El precómputo se lleva a cabo sobre un modelo iluminado por un mapa de entorno del que se derivan cálculos básicos de transferencia difusa sin sombras, con sombras (añadiendo al cómputo la visibilidad de un punto) y con interreflexiones.

$2B + 1$  valores a las series, es decir, que un orden 1 requiere 1 valor, un orden 2 requiere 4, un orden 3 requiere 9 y así sucesivamente. Cada nueva banda permite añadir más detalles al cálculo. Dado que el resultado ideal requeriría infinitos coeficientes, cada orden implica un menor o mayor grado de aproximación a este ideal o, en nuestro contexto, una supresión mayor o menor de las altas frecuencias. Por último, al proyectar los vectores correspondientes a la iluminación y a la función de transferencia en coeficientes SH, la integral se reduce a un cálculo vectorial simple, lo que simplifica y acelera extraordinariamente el proceso. En el método presentado por Sloan *et al.*, se recortan las frecuencias altas para simplificar el cálculo, lo que permite trabajar con un número reducido de coeficientes, reduciendo el cálculo a un producto vectorial punto de 9 a 25 vectores. Para concluir: los armónicos esféricos son funciones de base equivalentes a las series de Fourier para funciones expresadas en un dominio esférico con varias ventajas técnicas. De hecho, se ha utilizado en diferentes contextos, además de este, con resultados muy interesantes.

Se parte, por tanto, de la base de que la escena está rodeada por un mapa de entorno situado en el infinito. La iluminación en un punto dependerá, en principio, de la luz que llegue de este mapa. Los mapas utilizados en las publicaciones sobre este método suelen ser mapas HDR desplegados a partir de una *light probe* por lo que el método se emparenta con los métodos de HDR e IBL que hemos visto y continuaremos viendo en este capítulo, lo que le presta un interés adicional.

La respuesta de este punto se considera como función de transferencia de radiancia (*radiance transfer function*) que proyecta la radiancia incidente en radiancia de salida. Este cómputo inicial puede subdividirse en varios cómputos antidiendo a la mayor o menor simplificación de la ecuación de *rendering*. Puede tenerse en cuenta tan solo la difusión sin sombras, o bien la difusión con sombras añadiendo un factor de visibilidad o las sombras que el objeto arroja sobre sí mis-





mo o las interreflexiones. La figura 4.5 ilustra estas diferentes posibilidades.

Para capturar las muestras de radiancia en un punto se hacen 6 *renders* que corresponden a las 6 caras de un cubo que rodea al punto. Estos mapas se reproyectan sobre sus coeficientes SH. Para reducir el *aliasing* se supermuestran las imágenes correspondientes a estos mapas por un factor 2x.

Los pasos básicos del algoritmo son los siguientes:

- a) Precálculo de la radiancia global. Se computan los componentes de los armónicos esféricos (SH) de la función de transferencia para cada punto de la escena. En este precálculo solo se computa cómo la luz global afecta a las sombras que los objetos arrojan en torno a sí mismos o sobre sí mismos. Esto implica el cálculo de un gran número de integrales sobre los hemisferios centrados en cada punto, lo que se hace por métodos de Montecarlo. Los resultados se guardan en *buffers* de texturas.
- b) En tiempo real, la radiancia incidente se proyecta sobre las bases SH. El resultado es una función de transferencia (*transfer function*) dada en forma de vectores o matrices que se aplica al vector de iluminación. Esto supone rotar la representación de armónicos esféricos para alinearla con el sistema de referencia local y, a partir de ahí, calcular el producto entre los componentes de iluminación y la función de transferencia (el producto vectorial del vector iluminación y del vector de transferencia).
- c) Si el objeto es difuso, se utiliza un vector de transferencia (*transfer vector*), que se multiplica por el factor de reflectancia correspondiente para obtener una representación difusa de los diferentes objetos.
- d) Si el objeto es brillante (*glossy*) se utiliza una matriz de transferencia (*transfer matrix*) que ejecuta una convolución con las BRDF de los puntos considerados, seguida de una evaluación en la dirección *R* dependiente del punto de vista.

Estos resultados básicos se pueden combinar con otros cálculos. No entraré en los detalles pero se pueden consultar las referencias que doy más adelante. En cualquier caso, el uso de las SH permite simplificar el cálculo como se decía más arriba, considerando solo las frecuencias bajas. Con un pequeño aumento del número de coeficientes se pueden incluir también las reflexiones especulares simples (eliminando los realces especulares muy altos). Las superficies especulares se computan por medio de matrices en lugar de vectores.

Antes del artículo de Sloan ya se habían propuesto otros métodos de precomputación tales como *scene relighting* que precomputaba una solución de GI para cada luz. Véase el artículo original si interesan estas referencias.

Y con posterioridad también se han propuesto otros métodos. Puede encontrarse más información en un artículo sobre las tendencias generales en GPU-GI publicado por Tobias Ritschel, Carsten Dachsbacher, Thosten Grosch y Jan Kautz (Ritschel *et al.*, 2012), en el apartado 3.8. Entre estos métodos posteriores se ha ampliado las posibilidades representando la transferencia luminica mediante *wavelets* no lineales o mediante otras bases.

También ha habido muchos artículos y cursos posteriores que han explicado con mayor detalle este método. Véase, además de los artículos y cursos de Sloan, otros dos que cito en las referencias: Green, 2003 y Ramamoorthi, 2009.

### Ray tracing con la GPU

La posibilidad de incorporar *ray tracing* a la GPU es la línea de investigación que se ha seguido con mayor interés como una alternativa directa a la rasterización. De hecho, a menudo se plantea esta alternativa como disyuntiva: *ray tracing* frente a *rasterization* como ya hemos visto más arriba, en la introducción a esta sección. Las ventajas y desventajas de ambos métodos también se han resumido allí.

Lo primero que hay que recordar es que cuando hablamos, en general, de *ray tracing*





estamos hablando de procedimientos muy diferentes. Entre otras cosas porque hay métodos que utilizan casi exclusivamente *ray tracing* y hay métodos, como la mayoría de los que estamos describiendo, que lo utilizan parcialmente. En el capítulo 3 ya he descrito estas diferencias que están implicadas en el uso de diferentes términos: *ray casting*, *ray tracing* clásico o “estilo Whitted”, *ray tracing* de distribución o basado en métodos de Montecarlo y *ray tracing* de propósito general que puede ser parte integrante de otros métodos. También se han descrito los principales recursos de optimización basados en la construcción de estructuras envolventes de subdivisión espacial (*kd-trees*, *BVH*, *grids*, *BSP*, etc.).

En este apartado me centraré en los métodos que han intentado incorporar *ray tracing* a la GPU como método principal. En los demás apartados *ray tracing* aparece, de un modo u otro, como parte de una estrategia específica. La ubicación de este apartado en esta posición se debe a que estoy siguiendo aproximadamente un orden cronológico.

Uno de los primeros intentos de usar la GPU para *ray tracing* se llevó a cabo en 2002, por parte de N. Carr, J. Hall y J. Hart que presentaron un sistema híbrido, *the ray engine* que combinaba el procesamiento entre la CPU y la GPU. Pero la primera referencia importante es el artículo publicado en 2002 por Purcell, Buck, Mark y Hanrahan (véase Purcell *et al.*, 2002) pues en él se evalúan las posibilidades teóricas de incorporar diversas variantes de métodos de *ray tracing* a la GPU de un modo completo y sirvió de referencia para trabajos posteriores.

Estos autores comienzan por afirmar que lo más probable es que la *graphic pipeline* evolucione hacia su conversión en un procesador de flujos programable (*programmable stream processor*).

También parten de la base de que el estadio de rasterización correspondiente a los *pixel/fragments shaders* pronto incluirá cómputos en coma flotante y conjuntos de instrucciones más complejos. Pero en la mayoría de las GPU aún no es así. Un *vertex shader* in-

cluye del orden de 128 instrucciones en coma flotante de 4 vías. Un *pixel shader* incluye lo mismo pero con valores fijos. Es decir, a pesar de la importante novedad de que este estadio sea programable tiene varias limitaciones: a) un conjunto incompleto de instrucciones, b) formatos mayoritariamente fijos, no en coma flotante, c) un número limitado de registros de entrada y salida, d) un número limitado de texturas, e) un único valor de color que pueda ser registrado en cada paso, f) no hay derivaciones condicionales. Sin embargo, todas estas limitaciones pueden ser superadas con relativa facilidad como ya está ocurriendo y en la simulación propuesta todas las limitaciones anteriores se dan por resueltas. Y se considera que lo serán realmente con la evolución de las GPU como efectivamente está ocurriendo.

Por otro lado, las derivaciones condicionales, por ejemplo, son necesarias pues en una aplicación que incorpore *ray tracing* es necesario atravesar una determinada estructura y comprobar si hay intersecciones del rayo con la lista de triángulos que incluye dicha estructura. Y esto requiere ramificaciones condicionales (*branching*): *si* encuentras un triángulo *entonces* tal y tal y *si no* tal y cual. Pero esta limitación puede esquivarse mediante técnicas de pases múltiples. Se utilizan *stencils* (plantillas) para derivar el resultado en función de un test que devuelve *true* o *false*. Y solo se escriben en el *buffer* los valores de los diferentes pases que son *true*. Pese a todo, las implementaciones que se describen recomiendan evitar las derivaciones condicionales siempre que sea posible. En la simulación propuesta se evita el uso de derivaciones condicionales mediante técnicas de pases múltiples por medio de *streaming*.

El uso de *streaming* implica que los datos se leen mediante secuencias de flujos de elementos. La figura 4.6 ilustra la idea básica. Cada píxel corresponde a un elemento de flujo, y cada elemento de flujo es un tipo de dato que implica operaciones similares. Al representar un elemento de imagen se está activando un *shader* programable, es decir,



se está ejecutando un núcleo (*kernel*) del procesador de flujos. Los datos de entrada para ese núcleo se extraen de los registros de textura correspondientes a ese píxel. Y el resultado se escribirá en el *frame buffer*. Se parte de la base de que todo esto se llevará a cabo en GPU que permitan utilizar datos en coma flotante tanto para los registros de textura como para el *frame buffer*. Cada rayo corresponde a un elemento del flujo y cada operación sobre un rayo se expresa como un pequeño *kernel* realizable por un *shader*.

La reformulación de los algoritmos de *ray tracing* para que puedan ser procesados por la GPU mediante *streaming* requiere los siguientes cambios: a) La geometría de la escena debe representarse mediante una lista de triángulos guardados en la estructura de aceleración de hardware; b) La estructura de datos debe quedar integrada en algún tipo conocido. Hay varios tipos posibles: jerarquías de volúmenes envolventes, árboles bsp, árboles *kd*, *octrees*, retículas uniformes, adaptativas, jerárquicas, etc. Los autores escogen como referencia de partida retículas uniformes (*uniform grids*) pues consideran que los experimentos llevados a cabo con los diferentes tipos no demuestran que un tipo sea mucho más eficiente que otro y las uniformes son más fáciles de construir en la GPU.

El algoritmo se estructura en cuatro núcleos (*kernels*): a) generación de rayos desde la cámara (*eye ray generation*) con lo que se genera un rayo por cada píxel de la imagen final; b) travesía de la retícula (*grid traversal*): los rayos recorren la escena, que está estructurada como una textura 3D, hasta que encuentran un *voxel* que contiene triángulos, en cuyo caso pasa los datos al siguiente *kernel*; c) intersección de los rayos con los triángulos (*ray-triangle intersection*): las intersecciones computan el color correspondiente a la superficie de intersección; d) *shading*: se generan sombras o rayos secundarios que se devuelven al *kernel* de la estructura transversal. Véase la figura 4.6.

Estos *kernels* se implementan como *pixel/fragment shaders*. Y se combinan con texturas proyectadas sobre la pantalla de salida. El

*stencil buffer* controla qué fragmentos corresponden a texturas y a estado de rayos resultantes del proceso de los *kernels*.

El *shading kernel* evalúa el color correspondiente a cada punto de intersección. Los datos utilizados se almacenan en memoria de modo similar a los datos geométricos de los triángulos procesados. Un conjunto de tres texturas RGB, con 32 bits por canal, se utiliza para registrar la normal y los colores de los vértices de cada triángulo. Se pueden escoger entre diferentes tipos de procesamiento. Los autores describen los cuatro tipos siguientes:

*Ray caster*. Devuelve el color del píxel del triángulo que encuentra en la primera intersección. No se generan nuevos rayos. Es la aplicación básica.

*Whitted ray tracer*. Por cada intersección se generan nuevos rayos que se reenvían a la escena. La dirección depende de las propiedades de reflexión y refracción de la superficie encontrada.

*Path tracer*. Por cada intersección se envían rayos en direcciones aleatorias hasta que encuentran una luz. Se genera un *path* por cada muestra y cada *path* incluye 2 rebotes (*bounces*). Emula el *Arnold Renderer*

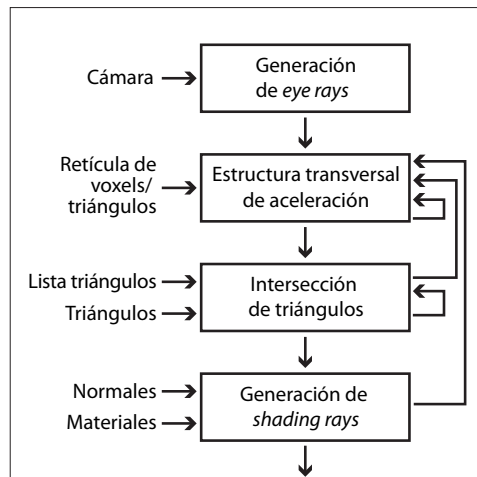


Figura 4.6 Streaming ray tracer. Adaptado de Purcell et al., 2002.



(véanse los resúmenes sobre motores de *render* del capítulo 6 o Fajardo, 2001).

*Shadow caster*. Simulan un renderizador híbrido que utiliza la *pipeline* estándar para calcular las superficies ocultas en el primer pase y luego usa un algoritmo de *ray tracing* para calcular las sombras. Es una buena alternativa a los métodos tradicionales de *shadow map* y *shadow volumes*. Genera rayos de sombra para cada luz y añade la contribución correspondiente a la imagen final si no se encuentran superficies que bloqueen el rayo en su camino.

Según los tipos, pueden generarse rayos secundarios para obtener sombras, reflexiones, refracciones o rayos aleatorios. A cada rayo se le puede asignar un peso diferente para que su contribución a la imagen final sea mayor o menor.

En el apartado 4.2 del artículo citado de Purcell *et al.*, los autores comparan la eficiencia de hacer múltiples pases en lugar de ramificaciones condicionales. Pese a que lo primero requiere del orden de 1.000 a 3.000 pases en su simulación, consideran que puede ser eficiente, pues, aunque la estructura de ramificaciones requiere muchos menos pases también requiere computar más objetos. Los apartados finales se extienden en consideraciones diversas sobre la eficacia del método y la comparación con otras investigaciones previas que consideran menos eficientes.

En los años posteriores se publicaron bastantes artículos en que se discutía principalmente la eficiencia de las estructuras de optimización. Doy alguna referencia adicional tan solo para dar idea de cómo ha evolucionado recientemente esta línea de investigación.

En Wald y Havran (2006) se discuten con detalle las diversas variedades de estructuras que existen para *ray tracing*: *octrees*, BVH (*bounding volume hierarchies*), retículas (*grids*) de diferentes tipos, uniformes o no uniformes, *kd-trees*, etc. De estas, los autores argumentan que los *kd-trees* son más eficientes. Pero señalan que no se ha tenido en cuenta suficientemente el coste computacio-

nal de construirlas y, teniendo en cuenta todos los factores, proponen un algoritmo que, entre otras cosas los construye en  $O(N \log N)$ , el límite teórico inferior considerado para este tipo de estructura.

En Zhou *et al.* (2008) se describe un método más eficiente de construir estructuras de tipo *kd-tree*. Y en un artículo posterior de R. Wang, K. Zhou, M. Pan y H. Bao (Wang *et al.*, 2009) se adopta este método de construcción para un método más complejo basado en *ray tracing*, *photon mapping* y *parallel cluster* que integra luz indirecta con múltiples rebotes, reflexiones dispersas, cáusticas y un número arbitrario de itinerarios especulares. Consideran que los métodos de construcción de *kd-trees* recientes, tales como los propuestos por (Zhou *et al.*, 2008), suponen importantes pasos para conseguir una adecuada implementación de las técnicas de *ray tracing* pero que la mayor dificultad reside en el último pase, *final gather*. Esto es debido a que se necesita acceder a los datos almacenados pues se requiere un procesamiento secuencial de los puntos muestreados y una actualización frecuente, que implica el acceso repetido a la estructura espacial en que se almacenan estos puntos. Volveremos a este aspecto más adelante.

### *Photon mapping*

*Photon mapping* con la GPU se implementó por primera vez a partir de un artículo de Purcell *et al.* (2003). Utilizaron una retícula uniforme para almacenar los fotones y un radio fijo. Esto supone una limitación del algoritmo original de Jensen que utilizaba un *kd-tree* y podía adaptarse a lo que interesara al usuario, eligiendo, por ejemplo, un radio más pequeño que generaba más ruido pero conservaba los detalles o un radio mayor que disminuía el ruido pero con más desenfoque y pérdida de detalle. La elección de una estructura más rígida impide jugar con estas variables pero posibilita construir la estructura directamente en la GPU. Además, utilizaban un algoritmo de búsqueda que mejoraba el resultado per-



mitiendo un crecimiento del radio mediante el examen de grupos de celdas concéntricas, en torno al punto analizado. Para la representación se utiliza un *stochastic ray tracer* basado en un *fragment shader*, de modo similar al utilizado por Purcell *et al.*, en el artículo previo de 2002. Y también utilizaban *ray tracing* recursivo para las reflexiones especulares y las refracciones y *ray tracing* distribuido para las sombras suaves de luces de área. También podían incluir la generación de cáusticas.

Buena parte del artículo está dedicado a describir los métodos de construcción de la estructura y los métodos de búsqueda. Dan dos métodos principales. El primero se procesa desde el *pixel/fragment shader* y utiliza un método denominado *bitonic merge* cuya efectividad es de  $O(\log 2n)$ . El segundo se procesa desde el *vertex shader* en un único pase, limitando el número máximo de fotones que se pueden escribir en una determinada celda y utilizando el *stencil buffer* para dirigir los fotones a su posición. Remito al lector interesado en estos temas, que se salen del marco de este libro, al artículo original.

Los artículos sobre construcción de estructuras de optimización, en particular el artículo de (Zhou *et al.*, 2008) sobre construcción de *kd-trees* que he citado al final del apartado anterior, también se utilizaron con éxito en variantes de *ray tracing* para aplicarlos a *photon mapping*.

En un artículo posterior de otros investigadores (McGuire y Luebke, 2009) se propone un método híbrido que reenvía los pasos más costosos de la computación a la GPU, rasterizando un mapa de rebotes dado en el espacio de la luz y los subsecuentes pasos se dejan para la CPU. Utilizan un método que denominan *Image Space Photon Mapping* (ISPM) para calcular la iluminación indirecta mediante un rastreo de una lista de *photon volumes*. Cada volumen delimita un *filter kernel* basado en la probabilidad estimada *a priori* para cada recorrido (*photon path*). Estos dos pasos sacan partido del hecho de que los segmentos iniciales de un *path* que

comienza en la luz y de los finales que terminan en la cámara, comparten un mismo centro de proyección. Dicho de otro modo: a) el segmento inicial, desde el emisor hasta el primer rebote es el que incluye más fotones y resulta más costoso de computar; b) el segmento final, que llega a la cámara, es igualmente costoso pues debe sumar todos los fotones que llegan a un determinado píxel; c) estos dos segmentos, inicial y final, tienen un centro de proyección único y pueden, por tanto, ser computados por medio de *rasterization*. Los autores introducen el concepto de *bounce map*, que acelera el paso inicial, y de *photon volume*, que acelera la estimación de radiancia del rebote (*bounce*) final. Las principales ventajas del método son que es muy eficiente y también que es fácil de desarrollar pues el código es bastante más simple y más compacto que otros métodos.

Los pasos del algoritmo son los siguientes: a) La escena se rastrea desde la cámara/ojo para crear un espacio de imagen o pantalla (*screen space*) que se almacena en un *G-buffer*; b) Para cada luz se rastrea la escena y el resultado se almacena en un *bounce map*. Este mapa es similar a un *shadow map* pero la información de cada píxel del mapa incluye la posición, energía y dirección del punto de superficie más cercano. Los fotones absorbidos tienen energía cero y dirección indefinida; c) Se computan los fotones remanente mediante un procedimiento clásico de *ray tracing* desarrollado en el espacio global. Este paso se lleva a cabo desde la CPU; d) Los fotones finales se envían al *screen space*, invocando la iluminación almacenada en el *G-buffer*, a partir del rastreo de *photon volumes*. Para cada volumen se lleva a cabo una operación de convolución mediante un filtro gaussiano, para reconstruir la iluminación indirecta global. La mayoría de los fotones no son tenidos en cuenta, lo que hace este último paso más eficaz pues no requiere la construcción de un *kd-tree*.

Dado que, como ya hemos visto, *photon mapping* es una técnica que funciona mejor en combinación con *irradiance cache* o *final*



*gather*, la mayoría de las investigaciones se han centrado en buscar alternativas para estas técnicas que se comentan en otro apartado, más adelante.

### *Ambient occlusion*

A pesar de sus limitaciones, la oclusión ambiental, *ambient occlusion*, es un método que se presta bien a ser precomputado debido a que es independiente de la iluminación. En Pharr (2004) se presenta un método sencillo de precomputación de la oclusión ambiental por medio de recursos propios de la GPU, rodeando la escena de luces virtuales (entre 128 y 1.024 en su propuesta) y utilizando *shadow maps* para calcular las sombras y, a partir de ahí, derivar la oclusión ambiental de un punto como una media ponderada de las sombras debidas a las diferentes luces.

Pero la referencia principal se basa en un artículo de Michael Bunnell (Bunnell, 2005) en que se describe un método eficaz y directo de implementación de *ambient occlusion* en la GPU por medio de un método de discretización de la escena a partir de una jerarquía de discos. En este método, el primer paso es convertir todos los datos poligonales en *surfels*, elementos de superficie, similares a los utilizados por otros autores que ya he citado. Cada elemento de superficie se define como un disco orientado con una posición, un vector normal y un área determinada. Estos elementos tienen una cara anterior y una posterior: la luz es emitida y reflejada desde la cara anterior y es transmitida y arroja sombras desde la cara posterior. Por cada vértice de la malla original se crea uno de estos elementos y, dado que el registro de los vértices ya incluye la posición y la normal, solo se necesita calcular el área. Para ello se toma la suma de un tercio del área de los triángulos que comparten el vértice o de un cuarto si se trata de *quads*. Los elementos se guardan en un mapa de textura que se enviará a un *fragment/pixel shader*.

Como ya hemos visto, el procedimiento general de *ambient occlusion* implica calcular el

valor de accesibilidad, que es el porcentaje de hemisferio sobre cada punto de superficie que no está ocluido por la geometría. Y, como también hemos visto, resulta útil calcular también la dirección de menor oclusión lo que se conoce corrientemente como el cálculo basado en la *bent normal*. El valor de accesibilidad se calcula para cada elemento como 1 menos la cantidad que los otros elementos hacen sombra sobre el elemento. El elemento que recibe sombra es el receptor y el que la arroja, el emisor. A partir de aquí se puede llevar a cabo el cálculo (el lector interesado puede ver las ecuaciones correspondientes en el apartado 14.2 del artículo de Bunnell).

Estos valores de accesibilidad se calculan en dos pases. En el primer pase se aproxima el cálculo sumando los ángulos sólidos correspondientes a cada elemento y restando el resultado de 1. Después de este primer pase algunos elementos serán demasiado oscuros pues la sombra arrojada por elementos en sombra también se suma. En el segundo pase se reduce este factor. En subsecuentes pases se reduce aún más. El cálculo de la *bent normal* se hace durante el segundo pase y los resultados de los pases sucesivos se graban en un mapa de texturas.

En el artículo se incluyen una serie de recursos técnicos para acelerar el cálculo mediante simplificaciones diversas, principalmente agrupando elementos similares y haciendo que el algoritmo atravesase jerarquías de elementos en lugar de elementos individuales.

Se puede hacer más realista el resultado añadiendo luz indirecta (un solo rebote) y luces de área y luz directa. También se calcula la transferencia de luz entre elementos mediante el término geométrico del factor de forma, disco a disco, de la ecuación utilizada. No se tiene en cuenta el factor de visibilidad sino que se utilizan técnicas de sombreado. A partir de aquí se ejecuta un pase del *radiance-transfer shader* para calcular la cantidad máxima de luz reflejada o emitida. Y luego un *shadow pass* que substrahe la luz total de cada elemento.





En trabajos posteriores se ha partido de este método pero procurando mejorar su eficacia de diversos modos. En Shanmugam y Arikan (2007) se propone un método de pases múltiples que separa el proceso de cálculo de *ambient occlusion* en dos fases principales: una que tiene en cuenta la oclusión causada por detalles cercanos, a menudo debida a zonas de alta frecuencia, y otra que procesa la oclusión causada por superficies distantes, a menudo debida a zonas de transiciones suaves, de baja frecuencia. Para la primera se utilizan métodos que operan en el espacio imagen y para la segunda, métodos que sacan partido de los recursos de las GPU para procesar versiones de baja resolución de las superficies distantes. Una ventaja importante del método es que es en buena parte independiente de la complejidad de la escena. El procesamiento local, de detalles cercanos, se lleva a cabo solo para los puntos visibles desde la cámara, registrados en su *z-buffer*. Estos valores se combinan con los datos dados por los registros del *buffer* de normales con lo que se sabe la orientación de los puntos en el espacio global correspondiente a esos puntos en el espacio imagen. Los autores denominan a ese *buffer* combinado el *ND buffer* (*normals/depth buffer*). Cada píxel de este *buffer* corresponde a una muestra de alguna superficie en el espacio global por lo que se puede usar como un *occluder* para otros píxeles. Para ello, se re proyecta como una esfera sobre la escena. Los autores prefieren utilizar una esfera en lugar de un disco, como Bunnell, pues la esfera previene posibles discontinuidades (cambios bruscos de la normal) dado que es simétrica y no está ligada a una orientación determinada. El procesamiento global, de superficies distantes, se lleva a cabo también por medio de esferas, en lugar de discos, por razones similares. Y se utiliza también un *ND buffer* como en el caso anterior. Por añadidura, se utilizan límites en el procesamiento, de tal modo que los valores registrados que caen por debajo de los límites dados se eliminan. Dicho de otro modo, al aumentar la distancia  $d$  correspondiente a un punto y una esfera, el radio de esta esfera

disminuye y la oclusión ambiental correspondiente a esta esfera disminuye igualmente hasta un valor despreciable. Por último, el procesamiento de los puntos lejanos se lleva al *fragment shader* por medio de una proyección plana de las esferas correspondientes, que se enviará al *buffer* de salida para que sea fusionado con otros valores. Todos estos procesos pueden desarrollarse en paralelo por lo que la eficiencia del método es considerable. No es un método exacto pero los errores no son apreciables y los resultados mostrados en el artículo son bastante convincentes.

Otra variante más reciente es debida a Hoberock (2013). El autor de artículo parte del de Bunnell y muestra algunos problemas que presenta en la práctica y propone algunos cambios para hacer más eficiente el algoritmo. Concretamente, dice que el método de

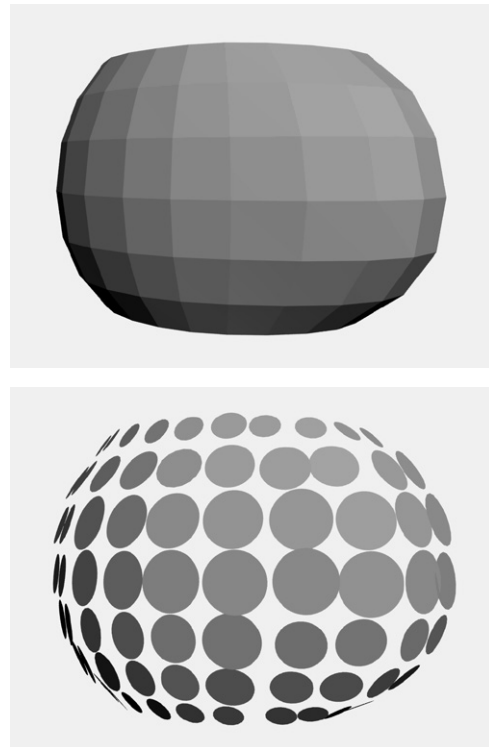


Figura 4.7 Ambient occlusion sobre la GPU. Conversión de una malla poligonal a surfels (adaptado de Bunnell, 2005).



representar una malla poligonal por medio de una serie de discos asociados a los vértices y computar la oclusión de cada disco por la sombra de los otros por medio de un factor de forma, no tiene en cuenta la visibilidad, pues los discos que están oscurecidos por otros no contribuirían con ninguna sombra. El algoritmo aproxima la visibilidad por medio de la oclusión: computa la oclusión de cada disco por otros y la almacena en el disco, luego, cuando se evalúa la sombra de cada disco se multiplica por la oclusión. De este modo se hace converger el resultado hacia una solución correcta sin necesidad de utilizar *ray tracing*.

El método de Bunnell utilizaba métodos más simples para las superficies distantes, creando agregados progresivos que se estructuraban en un árbol jerárquico de discos, pues la suma de cada interacción disco a disco sería muy costosa. De este modo, se pasaría de un recorrido que supondría  $O(n^2)$  a un recorrido que sería  $O(n \log n)$ , más eficiente. El método es adecuado para superficies suaves pero no tanto para superficies que presentan cambios abruptos y, en particular, para los puntos de contacto entre superficies en donde aparecen zonas de altas frecuencias que no pueden ser reconstruidas adecuadamente por medios de interpolación. Esto da lugar a dos tipos principales de artefactos: en forma de disco, que son debidos a discontinuidades en la transición de discos más toscos a discos más simples, y artefactos en forma de puntos (*pinching artifacts*) cuando se computan puntos que están muy cerca de discos.

Para solucionar estos problemas Hobbieck propone lo siguiente. Por lo que respecta a las discontinuidades, en forma de disco, de las sombras, define una zona alrededor de la frontera de transición en donde la oclusión suponga una interpolación entre la contribución de un determinado disco y la de sus descendientes en la jerarquía. Por lo que respecta a los *pinching artifacts*, comienza por situar los detalles más finos de los discos. Pero, en lugar de situar estos detalles más finos en los vértices, se sitúan en los centroides de las caras poligonales. Esto elimina buena parte

del problema. Y, en el cálculo, se evalúan los factores de forma, no a partir de discos, sino a partir de los componentes triangulares de la cara poligonal considerada, de un modo más exacto. Es decir, la ecuación asume que toda la cara poligonal es visible.

### *Final gather e irradiance caching*

Como ya hemos visto, el método más general de FG (*final gather*) consiste en dos pases: uno que lleva a cabo un cálculo muy simplificado, por lo general a partir de *photon mapping* aunque esto no es estrictamente necesario, y otro que refina este resultado a partir de los resultados del primer pase.

Este segundo pase es el más lento pues requiere computar un gran número de rayos. Para poner un ejemplo: en una escena más o menos típica puede ser que en el primer pase se envíen del orden de 500.000 fotones desde la luz y un número similar de rayos desde la cámara. Pero esto hay que multiplicarlo por el número de rayos reenviados y, para un número de muestras de FG de, por ejemplo, 1.024, haría falta enviar del orden de 250 millones de rayos, es decir, aproximadamente 1.000 veces más. Para una velocidad de procesamiento (CPU) de 250.000 rayos por segundo esto supondría más de 16 minutos (frente a 1 segundo para los rayos enviados desde la cámara). Por esta razón la mayoría de los programas utilizan un número mucho más reducido de muestras y métodos de interpolación, aunque esto puede requerir cierto tiempo dedicado a ajustar los valores para reducir errores de muestreo.

Dado el interés y la efectividad de los métodos basados en FG, una de las vías de investigación más activas se ha dirigido a buscar medios efectivos de incorporar estos métodos a la GPU. Para ilustrar estas vías de investigación me referiré muy brevemente a cuatro trabajos, uno de 2005, otro de 2008, otro de 2009 y otro de 2011.

En Hachisuka (2005) se introdujo un método de aceleración con FG utilizando los recur-





sos de rasterización de la GPU, a diferencia de métodos previos, como los de Purcell *et al.* (2002) que se basaban en la reestructuración de todo el proceso corriente de *rasterization* para adaptarlo a los métodos de *ray tracing*. El principio de que parte el trabajo citado es, en lugar de adaptar los métodos clásicos de *ray tracing* a la GPU, sacar partido de los recursos propios de la GPU por medio de nuevos algoritmos. Una ventaja importante de este enfoque es que no se necesita construir estructuras complejas (como los *kd-trees* o las *uniform grids*) con todos los problemas que hemos visto que esto conlleva.

La idea básica es la siguiente. El sistema calcula la integral sobre el hemisferio para, en principio, todos los puntos. El proceso se activa y se puede detener en cualquier momento por el usuario, cuando la imagen ha adquirido suficiente calidad. No hay por tanto una selección de muestras previa. Para optimizar el proceso, en lugar de computar todas las direcciones que emanan de los puntos de las superficies, se seleccionan todas las direcciones paralelas y se incluyen en un grupo (lo que los autores denominan *parallel clustering* por oposición al clásico *hemispherical clustering*). Estos grupos se procesan simultáneamente utilizando rasterización en la proyección paralela. La dirección seleccionada se denomina la dirección global del rayo (*global ray direction*). La idea (que viene de un artículo

previo de Szirmay-Kalos y Purgathofer de 1998), se ilustra en la figura 4.8.

Un aspecto importante del algoritmo de Hachisuka es la utilización de capas de profundidad, *depth layers*. Una *depth layer* es un subconjunto de la geometría de la escena obtenida a partir del número de intersecciones que se encuentran a lo largo de una determinada dirección (véanse las figuras). Para obtener estas capas por medio de la GPU se requiere llevar a cabo múltiples pases de *rendering* mediante un método que los autores denominan *depth peeling* (y que se basa en otro artículo de Everitt de 2001 cuya finalidad era distinta, ordenar la geometría para corregir la fusión de canales alfa). Con este método, se lleva a cabo un *rendering* normal. Pero, en el segundo pase, se ignoran los fragmentos de la misma profundidad correspondiente al primer pase. Y así sucesivamente.

La implementación se basa, por tanto, en una serie de pases que se describen en la sección 38.4 del artículo citado y que básicamente consisten en la transferencia de los datos a una serie de *buffers* relacionados. El método computa de este modo la iluminación indirecta. La iluminación directa se calcula de modo independiente durante el segundo pase y se suma a la anterior. El cálculo de iluminación directa con la GPU es muy rápido. Por lo que respecta al *aliasing*, con estos métodos no supone un problema importante.

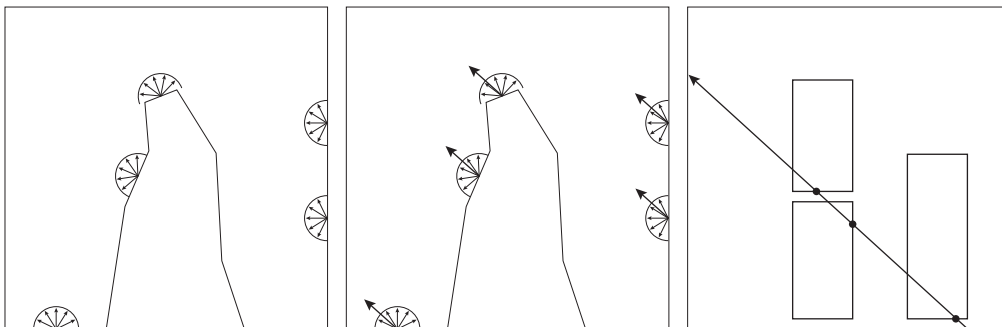


Figura 4.8 Final gather sobre la GPU. Tras el pase inicial en que se computa la irradiancia sobre un hemisferio centrado en los puntos muestreados, se seleccionan direcciones y se optimiza el proceso mediante capas de profundidad (*depth layers*) que permiten descartar partes de la escena que corresponden a puntos coincidentes (adaptado de Hachisuka, 2005).

En Christensen (2008) el procedimiento es, en un primer estadio, distribuir fotones por la escena, en segundo lugar, hacer una estimación de la densidad en cada punto para computar su iluminación y, por último, llevar a cabo un proceso de *final gather* para los píxeles de la imagen de salida.

La escena se representa mediante un proceso de precomputación por medio de *surfels* (un elemento de superficie dado por un punto y un disco orientado de radio variable que in-

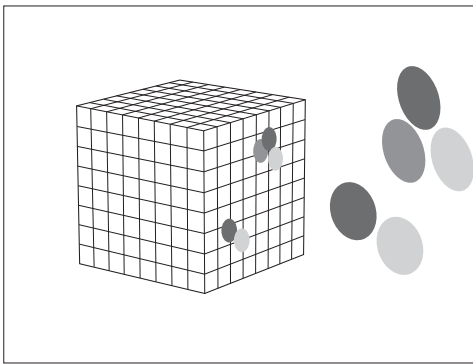


Figura 4.9 Final gather sobre la GPU. Rasterization de surfels sobre un minicubo que rodea al punto computado (adaptado de Christensen, 2008).

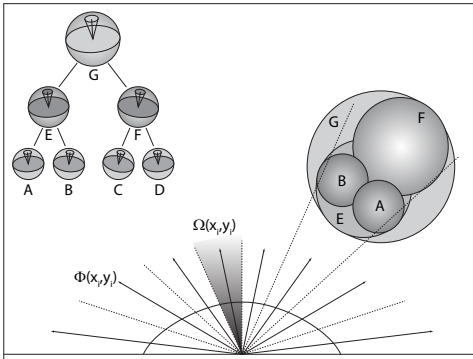


Figura 4.10 Final gather sobre la GPU. Micro-rendering. Arriba a la izquierda se muestra la jerarquía de puntos. Abajo, cada píxel del micro-buffer corresponde a una dirección  $\Phi(x, y)$  que abarca un ángulo sólido  $\Omega(x, y)$ . Los nodos de la jerarquía se atraviesan y rasterizan de tal modo que se proyectan sobre un píxel del micro-buffer (adaptado de Reitschel et al., 2008).

corpora datos sobre sus propiedades materiales), una nube de puntos que se estructura por medio de *clusters* en un árbol octal (*oct-tree*). La energía de cada *cluster* se aproxima por medio de armónicos esféricos (SH).

Para calcular la iluminación indirecta se comienza por llevar a cabo una representación muy tosca de toda la escena desde cada punto, rasterizando la escena visible desde cada punto sobre las seis caras de un minicubo que rodea al punto. A partir de ahí se calcula la interacción entre superficies, la fuga de color (*color bleeding*) por medio de tres técnicas que dependen de la distancia y la relación entre *surfels*. Los discos cercanos y que se solapan se computan con *ray tracing* para que la precisión sea mayor, los discos situados a una distancia media y que no se solapan se rasterizan, y los discos muy lejanos se computan de modo aproximado a partir de los datos de los armónicos esféricos de los *clusters* correspondientes.

Los resultados son aproximados pero convincentes y muy rápidos. El método de Christensen se ha utilizado en varias películas (como *Piratas del Caribe* entre otras) de un modo bastante eficaz.

En Ritschel et al. (2009), se encuentra otra alternativa similar para aplicar *final gathering* sobre la GPU, también basada en rasterization, por medio de lo que los autores denominan *micro-rendering* y que se basa en el uso de *micro-buffers*. El método consiste, sumariamente, en llevar a cabo, para cada punto de localización del *final gather*, un *micro-render* que representa y almacena el entorno que rodea al punto en un registro especial de memoria, en un *micro-buffer*. Así se crea una estructura jerárquica de la escena que luego puede ser atravesada para computar la iluminación global y llevarla al espacio de imagen de salida. El procedimiento es similar en esencia a *final gather* pero aplicándolo a diferentes puntos y sacando así partido del paralelismo de la GPU.

Para descargar al sistema se utiliza *importance sampling*. Esto permite escalar el proceso y adaptarlo a las necesidades del usuario. Si se necesita un resultado de alta calidad



se generan representaciones para cada localización de la imagen. Pero si se necesitan resultados más rápidos puede efectuarse el mismo cálculo sobre un subconjunto de las localizaciones.

El procedimiento consiste en los pasos siguientes: a) Se genera, previamente, una estructura jerárquica, basada en una selección aleatoria pero ponderada, de puntos de las superficies de la escena que posibilite una representación adaptada al nivel de detalle requerido. Esta estructura se basa en esferas envolventes y los nodos terminales representan un elemento superficial por medio de un disco orientado con un radio determinado. Con este tipo de representación, que, como la de Christensen y otros autores suele denominarse, *point-based representation*, se evita el costoso procesamiento de triángulos; b) A partir de aquí se generan microrrepresentaciones (*micro-renderings*) de estos puntos que representen la escena desde su posición, en torno, en principio, a un hemisferio y, en consecuencia, que almacenen la radiancia enviada desde su entorno inmediato. La representación computa la radiancia incidente desde una dirección determinada y basándose en las propiedades del material (BRDF); c) Se incorpora a este proceso un algoritmo de *importance sampling* para evaluar adecuadamente las regiones de mayor interés y reducir o suprimir el peso de otras regiones; d) Se lleva a cabo un proceso de *final gathering* para obtener el resultado final. Todo el proceso se

aplica utilizando los recursos de paralelización de los núcleos CUDA.

En Maletz y Wang (2011) también se combina *final gather* con *importance sampling*, tomando como datos de entrada una nube de puntos que representan la geometría de la escena que recibe iluminación directa y que, luego, se proyectan (se “aplastan”, *splat*, en el original) los puntos en *micro-buffers* que almacenan el campo de radiancia de oclusión de cada *shading pixel*. Los puntos para ser proyectados se seleccionan basándose en su importancia, que se define como la contribución estimada de cada punto al *shading pixel*. Para cada punto seleccionado se calcula su tamaño de proyección o “aplastamiento” (*splat size*) de modo adaptativo basándose en su valor de importancia. Para calcular la función de importancia se subdividen los puntos de la escena en *clusters*. El método es simple y rápido y permite incorporar otros factores de importancia tales como *glossy reflection paths*. También incluye un método de muestreo adaptativo en el espacio de imagen que combina subdivisiones adaptativas con supermuestreo para preservar detalles.

En el contexto de lo que se denominan genéricamente métodos de *caching* e interpolación, también podrían situarse otros trabajos de interés que me limito a mencionar. El problema que todos abordan de uno u otro modo es que los métodos clásicos de *irradiance caching*

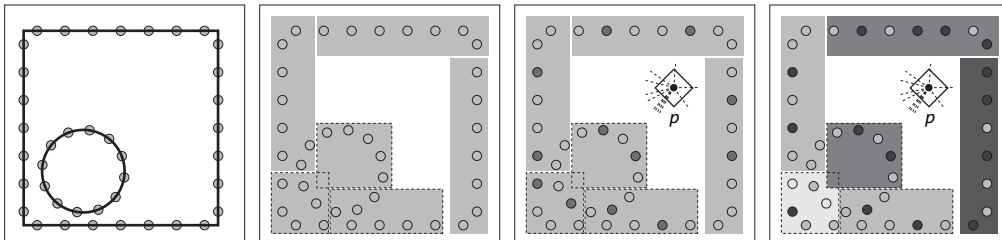


Figura 4.11 Final gather e importance sampling en la GPU (de Maletz y Wang, 2011): a) Se toman muestras de las superficies de la escena en varios puntos; b) Los puntos se agrupan en clusters; c) Desde el punto  $p$  se crea un microbuffer y se lanzan rayos aleatorios para estimar la importancia de los diferentes clusters; d) Se aplica una función de distribución de probabilidad para evaluar esta importancia (la intensidad de la trama ilustra la importancia de cada cluster) y se reproyectan las diferentes muestras sobre el microbuffer.



introducidos por Ward no pueden implementarse con facilidad en la GPU debido a que requieren la inserción secuencial de puntos de muestra de la escena, en una estructura de datos a la que se tiene que acceder repetidamente y que tiene que ser reconstruida durante el proceso de cálculo. No se adapta por tanto al procesamiento en paralelo que es la gran ventaja de la GPU.

Por ejemplo, *Radiance caching* es un método presentado por Krivánek y Gaudron, en 2005, que extiende el método tradicional de *irradiance caching* registrando la radiancia direccional mediante armónicos esféricos (SH). En otro trabajo de Arikan *et al.* (2005), se presenta una aproximación más rápida mediante la descomposición de los campos de radiancia en componentes cercanos y lejanos que se computan por separado para mejorar la eficiencia. Pero estos métodos siguen requiriendo la inserción secuencial. Y, por otro lado, hay métodos que hemos visto en apartados anteriores, como los *lightcuts* de Walter *et al.* (2005), que podrían incluirse en este apartado pues la finalidad y el procedimiento son similares. Otro tanto cabría decir de otros también mencionados más arriba, como los métodos de Hasan *et al.* (2007) para computar la iluminación por medio de matrices de luces puntuales.

## Aliasing y antialiasing

### Conceptos fundamentales

Cualquier imagen digital que se haya obtenido a partir de una imagen real está sometida a una limitación fundamental. Para representar la imagen real se habrán tomado una serie limitada de muestras. Si el número de estas muestras es insuficiente o si el número de valores asignados a cada muestra es también insuficiente, los resultados serán una pobre réplica de la imagen real. Ahora bien, esta limitación es inherente al procedimiento y, aún cuando el número de muestras y el número de valores sea “suficiente”, esta suficiencia solo lo será en determinadas condiciones de observación. A diferencia de

lo que ocurre con las imágenes reales, que varían a medida que nos acercamos o nos alejamos, pues lo que se ofrece a nuestra vista no es sino otro aspecto de una realidad que nunca podemos percibir en su totalidad, las imágenes sintéticas se degradan cuando nos acercamos o nos alejamos, pues no son sino un aspecto fosilizado de esta realidad cambiante.

La degradación se traduce en “alias”, elementos extraños que aparecen y desaparecen, pérdidas de información que quedan substituidas por información equívoca, que no corresponde a lo que debería corresponder. Para remediar esta degradación se necesita, en primer lugar, identificar los tipos principales de degradación y, en segundo lugar, buscar algún método de aliviar el problema.

En el primer artículo importante sobre este tema ya se identificaron estos tipos y se propusieron las soluciones adecuadas. A pesar de los años transcurridos desde que Franklin C. Crow publicó “Shadow Algorithms for Computer Graphics” (1977) conviene recordarlas. Las tres causas principales de *aliasing* que Crow citaba eran: a) los bordes de objetos o líneas en las superficies debidas a arrugas, cortes, etc.; b) los objetos muy pequeños y, c) las zonas con detalles muy finos.

En la terminología del procesamiento de imágenes se dice que estos problemas corresponden a zonas de “frecuencias altas”. Esta denominación es debida a que, en procesamiento de imágenes, todos los procedimientos se basan en la descomposición de una señal en sus componentes armónicos (funciones sinusoidales), en series de Fourier, así denominadas porque Jean-Baptiste Joseph Fourier (1768-1830) demostró, en una famosa obra sobre la teoría de la difusión del calor, publicada en 1822, que es posible esta descomposición en todos los casos que cumplan ciertas condiciones muy generales. Ahora bien, en el límite, un borde nítido, un corte brusco requeriría infinitas series sinusoidales para su análisis y se aproximaría mediante componentes armónicos de muy alta



frecuencia. En el libro sobre simulación de materiales me he extendido algo más sobre este tema y también sobre otros aspectos de la teoría de las señales que están implicados en los procedimientos de *antialiasing*.

Para remediar estas tres causas principales de *aliasing*, en el mismo artículo se proponían tres soluciones que siguen siendo válidas, si bien las dos primeras son bastante obvias y la más importante es la tercera:

a) Aumentar la resolución. Esto implica aumentar el número de muestras y, por tanto, reducir la posibilidad de que se pierda información por muestreo insuficiente. El inconveniente es que con esto se aumenta el coste de computación, por lo que habrá que ponderar muy bien este aumento. Una regla práctica que puede ser útil es tomar

como referencia las líneas más delgadas que vayan a aparecer en la imagen y comprobar que estas líneas quedarían cubiertas al menos por 3 píxeles. Pero este cálculo puede dar resultados prohibitivos en muchos casos.

b) Desenfocar los bordes de los objetos o las líneas internas para disimular el efecto de escalonamiento y de discontinuidad. Hay algoritmos simples, integrados en las tarjetas gráficas que añaden un píxel a cada lado con intensidad media o ponderada en función de la cobertura, entre el color del frente y el del fondo. El inconveniente de este recurso es que equivale a perder calidad y contraste si bien el remedio en este caso es mejor, casi siempre, que la enfermedad.

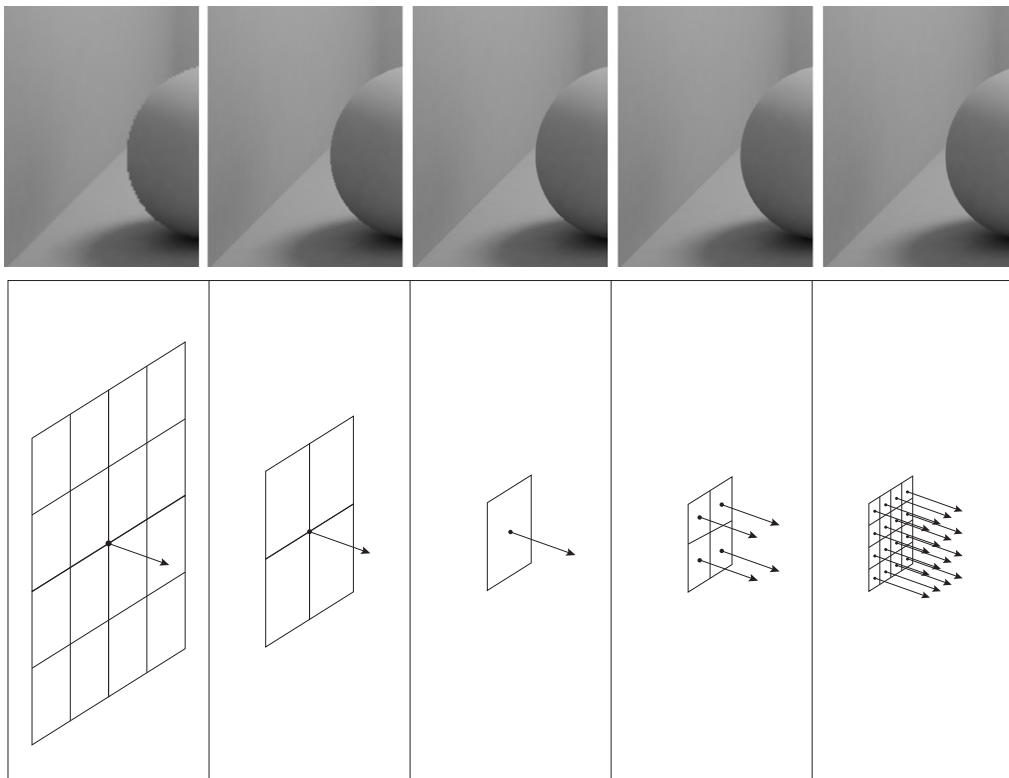


Figura 4.12 Submuestreo y supermuestreo: a) 1/16 (1 rayo por cada 16 píxeles), b) 1/4 (1 rayo cada 4 píxeles), c) 1 (1 rayo por píxel), d) 4 (4 rayos por píxel), e) 16 (4x4 rayos por píxel)



- c) Supermuestrear la imagen. Esta técnica, más compleja y con más variantes, se describe con algo más de detalle en el apartado siguiente. Me remito de nuevo al libro sobre simulación de materiales (capítulo 3, Recursos) para más detalles sobre estos temas.

### Supermuestreo y submuestreo. Muestreo adaptativo

Las dos técnicas más utilizadas en los programas de cálculo de iluminación son el supermuestreo (*supersampling*) y el submuestreo (*undersampling*). La primera se utiliza para mejorar la calidad del resultado y la segunda, para acelerar el cálculo.

El supermuestreo consiste en tomar, virtualmente, más muestras de las correspondientes a la resolución de salida y aplicar a esta imagen ampliada los cálculos de filtrado o interpolación para obtener, de este modo, resultados de salida mejor relacionados con la información de la imagen. Esto implica varias posibilidades de distribuir las muestras y ponderar los resultados. Una posibilidad es tomar muestras adicionales en las cuatro esquinas de cada píxel en lugar de su centro. Otra posibilidad es subdividir sistemáticamente los píxeles, lo que nos dará, para cada píxel, 4, 16, 64 píxeles, etc. Otra posibilidad es distribuir las muestras adicionales de modo aleatorio, lo que tiene la ventaja con respecto a las anteriores de que reduce los efectos de repetición causados por tramas regulares.

La ponderación de las muestras, por otro lado, también implica diversas alternativas. Podemos tomar valores iguales, lo que equivaldría a envolver el píxel con un prisma de igual altura. O bien tomar valores ponderados en función de la distancia al centro, lo que equivaldría a envolver el píxel con una superficie cuyo perfil fuese una curva adecuada que diera mayor peso a la región central y menos a la periferia. Variantes de todas estas técnicas están incorporadas a las tarjetas gráficas con variantes técnicas en

las que no vamos a entrar y que también se amplían en el capítulo mencionado del libro sobre simulación de materiales.

El submuestreo consiste en lo contrario, tomar menos muestras que las correspondientes a la resolución de salida. Un recurso corriente, análogo por el extremo opuesto al de supermuestreo por subdivisión sistemática, es especificar números negativos o fraccionarios que indicarían en este caso el número de píxeles que se agrupan para tomar una única muestra. Una especificación como  $-4$  o bien como  $1/4$  indicará que se va a tomar una muestra por cada 4 píxeles. Y una de  $-16$  o de  $1/16$  que se va a tomar una muestra por cada 16 píxeles, tal como se puede apreciar en la figura 4.12.

Por añadidura, estas configuraciones se pueden ligar a lo que se denomina muestreo adaptativo (*adaptive sampling*). El programa llevará a cabo un análisis previo de la escena detectando los lugares donde se producen transiciones bruscas. En estos lugares se tomarán más muestras que en aquellos en los que las transiciones son suaves, pues hay mayores probabilidades de que se pase por alto algún detalle significativo.

Estas especificaciones, presentadas de forma genérica, son las que se presentan en muchos programas. Por ejemplo, en mental ray, los parámetros de *antialiasing* se establecen a partir de configuraciones predefinidas que se basan exactamente en estos parámetros. Una configuración adecuada para pruebas se especifica como una especificación de calidad Min/Max de  $1/16$  a  $1$ , lo que quiere decir que se tomarán como mínimo 16 muestras por cada píxel en cualquier lugar y 1 muestra por píxel en lugares críticos. Y una configuración adecuada para la imagen final puede especificarse como, por ejemplo,  $1$  a  $4$ , lo que quiere decir que se tomará en cualquier caso 1 muestra por píxel o 4 en puntos críticos. El rango va de  $1/64$  como valores mínimos de submuestreo y  $64$  como valores máximos de supermuestreo. En el apartado siguiente amplío estas referencias elementales.





## Ejemplos de configuraciones básicas

Para comprender adecuadamente el método utilizado por la mayoría de los programas habría que referirse previamente al sistema de cálculo de la iluminación de la escena. Pero la mayoría utilizan algún método de trazado de rayos, sea de modo principal sea de modo complementario. Es decir, que se envían o se pueden enviar varios rayos por píxel hacia la escena. Partimos, por tanto, de métodos que implican la posibilidad de supermuestrear la escena. Estos rayos rastrean la escena de diversos modos, computando las interacciones y las desviaciones de los rayos, y acaban almacenando una información compleja pero que se traduce, en última instancia, en un color: el color que se asignará al píxel del que partieron.

En los cálculos finales, el número de rayos acabará, normalmente, por ser superior al número de píxeles (*supersampling*). Pero para pruebas previas puede ser inferior (*undersampling*). La configuración de muestreo de mental ray, como en otros programas, se basa en tres parámetros: el mínimo de muestras (*min samples*), el máximo de muestras (*max samples*) y el umbral de contraste (*contrast threshold*) que se descompone en tres valores (cuatro si se incluye un canal alfa), uno por cada color primario. El contraste puede especificarse de modo global o de modo diferenciado para los tres canales RGB. Esto permite hacer especificaciones más eficaces pues el peso no es el mismo para estas áreas. La luminosidad de una zona blanca de una imagen depende en un 59 % del verde (G), 30 % del rojo (R) y 11 % del azul (B). Esto quiere decir que los ajustes del canal R tendrán más repercusión que los del B. Aunque no es necesario preocuparse demasiado de estos matices conviene tenerlos presente aunque tan solo sea para entender por qué estos valores no se presentan ligados.

Estos tres parámetros, *min*, *max* y *contrast*, configuran un sistema de muestreo adaptativo característico: se trata de utilizar la menor cantidad posible de muestras, para conseguir

una imagen de calidad, en función de las características de la escena.

Hay otro parámetro adicional que conviene activar en determinados casos y que también se encuentra en este y otros programas de simulación: *jitter*, que introduce cierto grado de aleatoriedad en la distribución de las muestras, pues tiene poco efecto en el tiempo de cálculo y, en general, tiene un efecto positivo en el resultado, sobre todo si aparecen efectos de *moirée*.

En general, el proceso es el siguiente. En primer lugar, el motor de *render* toma el número de muestras especificadas por el primer parámetro, *min samples*. Los valores de estas muestras se comparan con los de las muestras adyacentes y, si la diferencia es mayor que la dada por el tercer parámetro, *contrast threshold*, se toman muestras adicionales hasta que la diferencia alcanza el valor umbral o hasta que se llega al límite definido por el segundo parámetro, *max samples*. Esto resulta en que se acabarán tomando más muestras en las zonas con más cambios perceptibles, con más detalles, que en las más uniformes. Valores altos de umbral harán que no se tomen muchas más muestras que las especificadas por el límite inferior y valores bajos de umbral harán que se tomen más muestras, hasta acercarse al límite superior.

El número de muestras por píxel viene dado por una ecuación cuyo sentido principal merece la pena recordar pues afecta exponencialmente al número de rayos. Si denominamos  $n$  al número de rayos visuales por píxel y  $m$  al número de muestras especificadas resulta que  $n = 2^{2m}$ . Así, para  $m = 0$ , tendremos que  $n = 1$  ( $2^0$ ). Para  $m = 1$  resultará  $n = 4$  ( $2^2$ ). Para  $m = 2$  resultará  $n = 16$  ( $2^4$ ). Para  $m = 3$  resultará  $n = 64$  ( $2^6$ ), etc. Cada paso cuadruplica el número de rayos, lo que equivale a subdividir el área del píxel en 4 partes iguales. Esto indica que habrá que manejar estos valores con precaución pues los tiempos de cálculo se pueden disparar con rapidez.

Si se utilizan valores negativos el efecto es el contrario. En este caso, estamos utilizando técnicas de submuestreo para acelerar el





cálculo. Un valor de -1 resultará en 1/4 rayos por píxel ( $2^{-2}$ ), es decir, un rayo por cada  $2 \times 2$  píxeles. Un valor de -2 resultará en 1/16 rayos por píxel ( $2^{-4}$ ), es decir, un rayo por cada  $4 \times 4$  píxeles.

Dadas todas estas posibilidades ¿qué configuración debe escogerse? La respuesta depende de dos contextos principales: la fase del proceso en que nos encontremos y el destino final de las imágenes. A lo primero se puede responder de un modo relativamente sencillo pues hay tres estadios característicos: a) pruebas rápidas, para comprobar si la iluminación es adecuada y los materiales se han asignado correctamente, b) estudios intermedios, por lo general análisis de detalle de zonas específicas, c) imágenes finales. A lo segundo no, pues depende del tipo de trabajo y del medio en que vaya a aparecer. Por ejemplo, una imagen que vaya a ser impresa a tamaño relativamente grande, pongamos que DIN A3, lo que supone una anchura efectiva de unos 35 cm o más (42 cm menos márgenes) requerirá un tamaño de salida de unos 4.200 píxeles (35 cm x 120 píxeles por cm). Pero si el destino de esta imagen es un monitor de 1.400 px de resolución bastará con esta cifra para el tamaño de salida. Los tiempos de cálculo diferirán considerablemente en los dos casos y esto afectará al proceso de trabajo pues se pueden forzar más o menos.

Para un caso intermedio (pongamos que una resolución de salida de unos 2.048 píxeles) podemos tener presentes las referencias siguientes. Para las pruebas rápidas pueden especificarse valores de contraste superiores a los predeterminados (0,05) que pueden ser de hasta 0,5. Valores inferiores a 0,05 difícilmente se apreciarán. Pero pocos usuarios se molestan en ajustar este parámetro para pruebas rápidas aunque convendría jugar al menos con los valores predeterminados almacenados en los *presets*. Los valores de muestreo pueden ser tan bajos como 1/64 o 1/16, es decir, valores de submuestreo. Pero esto dependerá del tipo de escena y convendrá aumentarlo si la escena incluye reflexio-

nes dispersas, sombras de área o efectos de profundidad de campo.

Para las pruebas finales no suele ser necesario superar los 64 rayos por píxel y 16 será suficiente en la mayoría de los casos.

Si *min samples* se iguala a *max samples* se anula el muestreo adaptativo y tendremos directamente *supersampling*, si el número de rayos es mayor que 1, o *undersampling*, si es menor que 1.

Aunque la descripción anterior está referida a mental ray, para simplificar la exposición, los conceptos de base y los tipos de valores son similares en otros programas de simulación. Otro tanto vale para lo que sigue.

### ***Unified sampling. Progressive rendering***

En las últimas versiones de mental ray se ha introducido otro método de configuración de *antialiasing* que resulta más cómodo de utilizar. De hecho, estos métodos estaban incorporados al programa desde 2010 aproximadamente (desde la versión 3.8 de mental ray) pero estaban ocultos pues no habían pasado las pruebas de verificación. Como todo lo que tenía que hacer un desarrollador para utilizarlos era eliminar un signo de comentario en los códigos internos accesibles de las bibliotecas incorporadas en el programa, desde entonces han aparecido varios scripts y plug-in que incorporaban estos métodos. Pero a partir de la versión 3.11 de mental ray (versión 2014 de 3ds Max y correlativas de Maya y Soft Image), están disponibles directamente en la configuración general a la que se accede desde la configuración de salida (*render setup*).

En *render setup / renderer/ sampling quality* encontraremos un nuevo grupo, *sampling mode*. Este nuevo grupo permite elegir entre tres modos: a) *unified / ray traced*, b) *classic / ray traced* (es el modo descrito en los apartados anteriores y que se basa en valores fijos de *min*, *max*), c) *rasterizer / scanline*. Es una versión actualizada y mejorada del modo clásico de *scanline*.

El primero, *unified*, es el nuevo modo. La diferencia con el clásico, basado en valores



fijos de *min*, *max* es que se puede jugar con un nuevo parámetro global, *quality*, predeterminado a 0,25 sobre un rango de 0,0 a 20,0. Y los valores predeterminados de *minimum* y *maximum* son 1 y 128 respectivamente (sobre un rango de 0,1 a 64,0 para *minimum* y de 1 a 100.000 para *maximum*). Pero solo tenemos que preocuparnos, en general, con establecer un valor global para la calidad lo que permite ajustes más finos y más sencillos.

Por otro lado, también está (relativamente disponible) la función de *progressive rendering*. Este nuevo modo de configuración computa y representa la imagen a plena resolución y la va refinando de modo incremental a diferencia del modo previo, en que la imagen se representa por paquetes (*buckets*) y no es plenamente visible hasta que no se han computado todos los paquetes. Esto permite contar con una apreciación global del resultado final y cancelar el proceso si los resultados que se van mostrando no son los deseados. También es más rápido que los métodos anteriores. Es el método utilizado implícitamente por el nuevo motor de rendering *iray*. Y es también el método utilizado desde hace muchos años por programas como V-Ray cuando se utiliza el modo de cálculo *light cache*.

El problema es que estas opciones no son accesibles directamente. Dos desarrolladores, Thorsten Hartmann y Artur Leao, han desarrollado varios *scripts* para sacar partido de estos recursos. Uno de ellos, *render optimizer*, puede ser particularmente útil pues incorpora múltiples variables que no están disponibles directamente.

Pero si no queremos depender del uso de *scripts* externos y todo lo que nos interesa es acceder a alguna de las variables principales, lo que se puede hacer es copiar en el Bloc de Notas de Windows lo siguiente y, cuando interese, pegarlo en *string options*. Si todo lo que nos interesa es activar *progressive rendering* con los valores predeterminados, bastará con copiar la primera línea. Y si queremos ajustar alguna de las variables cuya descripción doy a continuación, bastará copiar la línea o

líneas correspondientes. Solo doy las principales. Las demás pueden consultarse en la ayuda de mental ray pero la secuencia es similar: la palabra clave entre comillas seguida de un valor entre comillas si es una cadena y sin comillas si es un valor numérico.

```
"progressive" on
"progressive subsampling size" 4
"progressive max time" 20
"progressive error threshold" 0.02
```

Los parámetros principales disponibles con este recurso simple son los siguientes:

**Sampling size (8).** Con este parámetro se controla la resolución inicial que se irá refinando a medida que avance el proceso. El valor dado especifica el tamaño de los bloques en que se colocan inicialmente las muestras. Cuanto más alto sea este valor más tosca será la imagen inicial y más rápido se procesará. Valores de 0 o 1 desactivan este parámetro y un valor de 2 indicaría que se tomen muestras iniciales de 2 x 2 píxeles.

**Sampling mode.** Controla la apariencia de los píxeles submuestreados antes de que se representen. Tiene dos opciones: *detail* hace que los píxeles no procesados se representen con valores interpolados tomados de los píxeles vecinos, lo que se traduce en una apariencia inicial suavizada; *sparse* hace que los píxeles no procesados se representen negros, lo que se traduce en una apariencia inicial con bastante ruido.

**Sampling pattern.** Controla la secuencia con que se computan los píxeles. Tiene dos opciones: *scatter* hace que los píxeles se computen en un orden semialeatorio dentro de cada bloque; *linear* hace que los píxeles se computen en un orden basado en líneas sucesivas dentro de cada bloque.

**Sampling min samples y Sampling max samples** controlan el número mínimo y máximo de muestras por píxel del resultado final. Es adaptativo en función del umbral de error especificado (*error threshold*), que es un valor escalar simple. Si no se especifican estos valores el *render* se detendrá en función de este



umbral de error. Valores predeterminados característicos pueden ser 4, 100 (min, max)

**Max time** (sec). Especifica la duración máxima del proceso de *rendering*. El valor predeterminado en los *scripts* que lo utilizan suele ser de 10 segundos.

**Error threshold**. Especifica el umbral de error a partir del cual se detendrá el proceso. Si la calidad de la imagen queda por debajo de este umbral el *rendering* se detendrá automáticamente. Si el valor dado es 0,0 la calidad de referencia es perfecta y el proceso no se detendrá hasta que se cancele. El valor predeterminado es 0,05, que proporciona un resultado de calidad baja, para el modo general. y de 0,15 para *iray*.

### Filtros. Filtros usados en cálculos de iluminación

Un filtro, en procesamiento digital de imágenes, se puede definir como cualquier sistema que modifica una señal para suprimir o resaltar alguno de sus rasgos. En teoría de señales esto implica encontrar una función y una operación matemática adecuadas que puedan aplicarse automáticamente a la señal para modificarla de un modo característico. La operación matemática principal que se utilizan en filtros es la convolución. Una convolución es una combinación compleja de dos funciones que da lugar a una tercera función. El término viene del alemán *Faltung* (que se podría traducir como “pliegue” o “doblado”) y fue utilizado por Hilbert a principios del siglo xx. Norbert Wiener utilizó este término alemán en un libro de 1933 sobre la integral de Fourier y sus aplicaciones, a falta de una traducción conveniente. Pero pocos años después ganó aceptación el término *convolution* o “convolución” y en la actualidad es un término fundamental en la teoría del procesamiento de imágenes. El lector interesado encontrará amplia información sobre este concepto en internet. Para los fines de este capítulo nos bastará con recordar que es la herramienta matemática a partir de la cual es posible manipular las frecuencias de una

imagen de diversos modos y, más concretamente, utilizar todo tipo de “filtros” lo que, intuitivamente, se correspondería con “dejar pasar” ciertas características que nos importan y bloquear el paso de otras que no nos interesan.

El uso de filtros tiene más importancia en el caso de simulación de materiales y, de nuevo, me remito al libro sobre este tema donde lo he tratado algo más extensamente. A menudo interesa aplicar filtros a una simulación para corregir defectos derivados de cosas tales como reflejos o mapas de relieve (*bump maps*, *normal maps*). En los cálculos de iluminación se utilizan de modo más general para corregir defectos característicos derivados de un número de muestras insuficientes. Como ya hemos visto, esto se corrige principalmente mediante técnicas estándar de *antialiasing*, concretamente mediante *supersampling*. Pero puede haber casos en los que convenga utilizar métodos más específicos para prevenir determinados defectos. En este sentido, el uso de filtros puede verse como un desarrollo más especializado de las técnicas de *antialiasing* que hemos introducido en la sección anterior.

Pero también debe verse en el contexto de la integración de los cálculos de iluminación avanzada en la GPU. Los diversos intentos de sacar partido de la capacidad de procesamiento de la GPU han llevado a ensayar multitud de técnicas, entre ellas, como hemos visto, el uso de múltiples luces virtuales que simulan el comportamiento de la luz difusa indirecta. El principal problema que lleva aparejado este tipo de técnicas es que se crean discontinuidades características, debido al uso de elementos discretos, que requieren el uso de filtros especiales para reducir el ruido. O bien a utilizar variantes de técnicas clásicas de proyección de fotones o *path tracing* pero reduciendo el número de muestras para acelerar el cálculo, lo cual da lugar, de nuevo, a ruido, que puede ser reducido utilizando técnicas de filtrado especialmente adaptadas a los defectos característicos de este tipo de cálculos.



Una alternativa importante a la transformada de Fourier que ha surgido en relación con estos problemas es la *wavelet transform* (que se basa en trabajos del matemático húngaro Alfred Haar de 1909; el término fue propuesto en 1984 por Jean Morlet y Alex Grassmann). Lo que hace esta operación es proyectar la señal original sobre una serie de nuevas funciones básicas que se denominan *wavelets*. Estas funciones operan localmente y admiten diferentes resoluciones (diferentes muestreos sobre la señal original). Se ha utilizado mucho en sistemas de compresión (como jpeg2000) pero también en reconocimiento de bordes o patrones.

## Organización de la escena

Como ya he recordado en la introducción, no es posible separar nítidamente los procesos correspondientes a la simulación de materiales y los correspondientes a la simulación de la iluminación. Y, por añadidura, hay toda una serie de procedimientos implicados que no pertenecen ni a uno ni a otro campo. En este breve apartado resumiré lo principal que hay que tener en cuenta, tanto con respecto a lo que corresponde a estos últimos, los procedimientos comunes, como a lo mínimo que hay que tener presente sobre asignación de materiales, aunque nuestro análisis se vaya a concentrar en la iluminación.

## Aspectos generales

Para organizar una escena virtual se necesita, como mínimo, lo siguiente:

- a) Un modelo geométrico completo de la escena. Este modelo puede generarse con la misma aplicación con que se va a llevar a cabo el proceso de simulación o puede generarse en otro programa e importarlo a la aplicación.
- b) Una asignación de materiales a los diferentes objetos de la escena. Si lo que nos interesa es ajustar la calidad y distribución de la luz es recomendable, como paso previo, aplicar

a toda la escena un único material de color gris medio para que las diferencias de color y reflectancia de los materiales no interfieran con el juicio sobre la distribución de luz en la escena que estamos configurando.

- c) La preparación de una varias cámaras adecuadas para las vistas que queremos obtener de la escena.
- d) Un sistema de luces que, en general, consistirá en una luz primaria y una o varias luces secundarias que atenúen la dureza de las sombras de la primaria. Pero las variantes y ampliaciones de este esquema elemental son infinitas.
- e) Un sistema de cálculo que deberá configurarse de diferentes modos: para pruebas iniciales, para análisis de detalles y para la salida final.

Lo primero, el modelo geométrico, es muy importante pues las características geométricas del modelo afectan al resultado final. Es un tema que cae fuera de este libro pero sobre el que cabe recordar un par de cosas. El modelo debe estar bien formado, con vértices conectados en todas las caras (algo que no siempre ocurre) y con un número de facetas adecuado para simular superficies curvas. En determinados casos puede ser necesario utilizar objetos con diferente resolución o aumentar la densidad de la malla para objetos que vayan a ser procesados a corta distancia.

Lo segundo se trata en el libro sobre simulación de materiales al que ya me he referido. En el apartado siguiente doy algunas indicaciones adicionales sobre el uso de materiales de superposición y sobre el tipo de material más adecuado para la mayoría de los casos.

Lo tercero se trata en el apartado que sigue al del material de superposición.

Lo cuarto y quinto constituyen el tema propio de este libro.

## Material de superposición (*override material*)

Como decía anteriormente, para ajustar la configuración de iluminación es conveniente utilizar un material único, mate, de color



gris medio, para toda la escena, pues esto nos facilitará el trabajo de juzgar si la distribución de la iluminación es adecuada para que la escena no resulte demasiado oscura o demasiado clara, o le falte o le sobre contraste.

Para no tener que reasignar todos los materiales a todos los objetos hay dos soluciones características que dependerán de la aplicación utilizada. En algunos programas de aplicación puede especificarse, desde la configuración de salida, un material de sustitución (*material override*). Este material se aplica, en el momento de la representación a todos los objetos de la escena, substituyendo al material que estos objetos tuvieran asignado. Luego, basta con desactivar la asignación de este material para que todos los objetos recuperen su material original.

En 3ds Max, por ejemplo, todo lo que hay que hacer es abrir el *render setup* y, en la sección *processing / translator options*, en el grupo *material override*, activar esta opción y luego pulsar el botón junto a *material*, para escoger un tipo de material. Con esto se entrará en el navegador de materiales y mapas. Convendrá escoger el material *Arch&Design*, de mental ray que es la mejor opción para la gran mayoría de los casos, dejar los valores predeterminados y comprobar que los valores de reflectancia y transparencia son 0,0. Dar al color propio (*diffuse*) un valor RGB que corresponda a un gris medio. Eso es todo. Al hacer un *render* este material se superpondrá a todos los objetos. Para desactivarlo, todo lo que hay que hacer es volver a la sección mencionada del *render setup* y desmarcar la opción *enable*.

Si el programa de aplicación no cuenta con esta posibilidad lo más sencillo será guardar la escena con otro nombre, asignar a todos los objetos un material similar al anterior y, luego, si el programa lo permite, guardar o exportar las luces utilizadas y la configuración de salida. Si esto no fuera posible, anotar los valores correspondientes y replicarlos en el archivo original.

## Cámaras

Para preparar la imagen de salida hay que crear cámaras virtuales en una o varias posiciones. En la mayoría de los programas las cámaras se crean como un objeto más, que se sitúa en la escena y se puede mover y girar para que apunte en una determinada dirección. Una vez creada la cámara puede configurarse la salida (la configuración de *output* o de *render setup*, según los casos) para que muestre la vista de cámara. Hay dos tipos principales de cámaras: cámaras básicas y cámaras físicas.

Las *cámaras básicas* proyectan la escena en perspectiva sobre un plano virtual y prescindir de otras consideraciones. Esto quiere decir que, en estos casos, las únicas propiedades iniciales que nos importan son el campo de visión y el formato de salida. El campo de visión, o FOV por sus siglas en inglés (*field of view*) se controla modificando el ángulo de apertura de la cámara. El formato de salida se controla especificando unas determinadas dimensiones de anchura y altura de la imagen lo que, generalmente, se hace desde la configuración general de salida (*output* o *render setup*). En la mayoría de los programas de simulación (pero no en todos) también se puede activar algún parámetro especial para que el marco de salida se muestre en el visor de trabajo, de tal modo que se pueda componer adecuadamente la escena sin tener que representarla una y otra vez para comprobar el resultado.

Otras propiedades que se pueden encontrar en las cámaras virtuales básicas son: la activación de planos de recorte, la activación de ajustes de profundidad de campo y la activación de ajustes de rangos de entorno o rangos atmosféricos.

Los planos de recorte (*clipping planes*) permiten excluir parte de la geometría de la escena especificando un plano cercano (*near*), que suprime todo lo que hay entre él y la cámara, y otro plano lejano (*far*), que suprime todo lo que queda más allá de su ubicación. Es un recurso muy útil para ver el interior de



una escena sin necesidad de forzar excesivamente el ángulo de visión. Y así se obtiene un espacio, una pirámide truncada, que se corresponde exactamente con el *frustum* que he descrito antes y que es un componente estandarizado de la *rendering pipeline*.

La profundidad de campo o DOF por sus siglas en inglés (*depth of field*) está disponible, según los programas, como recurso propio de una cámara básica, como recurso propio de una cámara física (véase el apartado siguiente) o como recurso independiente que hay que cargar, sea desde la cámara, sea desde la configuración de salida, sea como efecto de postproducción que se superpone al resultado final. Según los programas, hay que especificar la distancia a partir de la cual se desenfocarán los objetos o señalar directamente objetos cercanos y lejanos que proporcionen este rango de desenfoque.

Los rangos de entorno (*environment ranges*) se encuentran en algunas aplicaciones como recurso adicional que puede asociarse a efectos atmosféricos tales como la simulación de niebla. De modo similar a los casos anteriores, implican especificar un rango, una distancia cercana y otra lejana, que definen el espesor sobre el que se aplicarán estos efectos.

Algunos programas incluyen también un operador adicional que permite rectificar las verticales, sea como parte de los controles de la cámara (en V-Ray, por ejemplo, se incluye el parámetro *vertical shift*) sea como recurso independiente que se superpone a las propiedades de la cámara (en 3ds Max, por ejemplo, se encuentra en el menú *modifiers / cameras / camera correction*). Es un recurso importante en arquitectura pues permite simular el procedimiento clásico de los fotógrafos antiguos, que utilizaban cámaras de fuelle con las que era posible desplazar el paralaje entre la lente y el negativo, algo que actualmente se lleva a cabo *a posteriori*, por medio de programas de manipulación de imágenes.

Las *cámaras físicas* incluyen todos los controles anteriores y otros ajustes adicionales para ajustar el control de exposición. Esto

es importante pues cuando se utilizan sistemas de iluminación avanzadas nos encontramos con rangos de valores muy altos que requieren este tipo de ajuste, igual que ocurre con las cámaras reales. Estos ajustes se dan generalmente por medio de parámetros iguales a los de las cámaras reales. En V-Ray, por ejemplo, nos encontramos con controles para ajustar la exposición por medio de los tres controles clásicos: apertura del diafragma (*f-stop*), velocidad de obturación (*shutter speed*) y velocidad de la película (*ISO*). Y, además, efectos de distorsión, simulación del viñeteado o controles adicionales para el balance de blancos.

En otros programas, como 3ds Max, el control de exposición se ajusta de modo independiente, como un control que se aplica a la salida de la imagen, con independencia de si esta salida se hace a través de una cámara o no. Pero los parámetros utilizados son los mismos si bien mental *ray* incluye la posibilidad de utilizar un único parámetro ponderado (EV, *exposure value*) que resulta algo más práctico que la combinación de aperturas de diafragma y velocidades de obturación.

El ajuste del control de exposición es un tema importante al que volveremos al final de este capítulo por lo que detengo aquí la explicación que retomaré más adelante.

En algunas cámaras físicas también se puede encontrar un efecto adicional denominado *bokeh* en el mundo de la fotografía. Este término es una palabra japonesa que se puede traducir por “desenfoque” pero implica algo más pues, cuando se desenfoca el fondo de una imagen para que resalte mejor el sujeto que está en primer plano, los objetos lejanos aparecen como manchas más o menos circulares. Pero estas manchas pueden tener diferentes formas por razones que están relacionadas con las láminas del diafragma. Una apertura que esté formada por cinco láminas creará formas ligeramente hexagonales mientras que una apertura con más láminas o menos desenfocada creará formas circulares. Estas formas pueden afectar notablemente a la calidad del fondo y muchos fotógrafos las ajustan para





que se adapte a lo que les interesa. Algunas cámaras virtuales incorporan la capacidad para crear este efecto y ajustar la forma y la proporción de estas manchas de desenfoque.

Otro valor que a veces se encuentra en las especificaciones del *output* o *render setup* es el *pixel aspect ratio*. Es un valor que está fijado en 1,0 (píxeles cuadrados) y que no hay ninguna razón para cambiar. Pero no está de más saber que hay dos formatos en los que los píxeles no son cuadrados: NTSC (*national television system committee*), el formato propio de la TV de Estados Unidos y buena parte de América Latina, que tiene una relación de aspecto de 0,9 y PAL (*phase alternating line*), el formato propio de la TV de la mayoría de los países de Europa y otros países, que tiene una relación de aspecto de 1,0 para vídeo de 768 x 576 y de 1,06667 para vídeo D-1 de 720 x 576. Esto es algo que conviene recordar en caso de que haya que trabajar con estos formatos por razones profesionales. Pero en general se puede pasar por alto y dejarlo en su valor predeterminado de 1,0.

## 4.2 Uso de imágenes HDR en cálculos de iluminación

### Imágenes HDR

En el cálculo de iluminación no intervienen, en principio, texturas o imágenes, por lo que todo lo relativo a estos temas se aborda en el libro sobre simulación de materiales.

Sin embargo, no es posible establecer una separación estricta. En muchas escenas intervienen fondos o mapas de bits que afectan al propio cálculo de iluminación. Un ejemplo es el de los mapas asociados a luces celestes, de tipo *skylight*: el mapa, aunque sea de tipo LDR (*low dynamic range*) puede afectar en mayor o menor grado a la distribución de la luz. Por otro lado, ya he mencionado que cualquier sistema de iluminación se verá alterado por las características de los materiales de la escena, por lo que el procedimiento

recomendado, indicado al final de la sección anterior, es configurar la escena con un material global único de color gris medio, para que los materiales no interfieran en el análisis de la distribución de la luz y, luego, reasignar los materiales y revisar los ajustes analizando los cambios, sutiles o no tan sutiles, que han surgido en la escena.

Sin embargo, hay un formato, HDR, que tiene una importancia fundamental pues está relacionado directamente con métodos específicos de cálculo.

El formato HDR ya se ha introducido en el capítulo 2 por lo que para recordar sus características básicas, la estructura interna, el número de valores que puede almacenar según los diferentes formatos, etc., me remito a este capítulo que el lector deberá revisar en paralelo a este.

En este capítulo se describen exclusivamente los métodos de obtención y edición de mapas HDR. Recordemos que también es importante tener claro para qué se van a utilizar estas imágenes. La finalidad principal, en nuestro caso, es obtener imágenes HDR que se puedan utilizar como mapas de entorno para simulación IBL. Esto quiere decir que no solo necesitamos imágenes HDR (que incorporan los rangos de valores adecuados) sino que necesitamos imágenes HDR que se desplieguen, a ser posible, en un panorama esférico y con una resolución y una calidad adecuada para la simulación que queremos llevar a cabo.

El modo de utilizar imágenes HDR para iluminar una escena se ha descrito genéricamente en el apartado sobre IBL del capítulo 3 y se retomará en el capítulo 6 con ejemplos de procedimientos y configuraciones concretas, incluyendo mayores precisiones sobre lo que queremos decir con una resolución y una calidad adecuada.

### Métodos generales de obtención de imágenes HDR

Hay tres métodos principales para obtener imágenes HDR: a) tomando series de fotografías con varias exposiciones y combinándolas





las mediante métodos especiales, *b*) creando imágenes sintéticas mediante un programa de modelado y simulación 3D y *c*) tomando fotografías directamente con cámaras especiales.

Naturalmente hay una cuarta posibilidad que no debe perderse de vista: obtener imágenes digitales a partir de terceros, sea descargando imágenes gratuitas, sea intercambiándolas, sea comprándolas. Pero no creo que esto se pueda calificar propiamente de “generar” imágenes HDR, por lo que me limito a recordarlo. Para obtener imágenes gratuitas HDR, aunque tan solo sea para hacer pruebas, la mejor recomendación es buscar por internet utilizando las palabras clave “hdr images” o “openexr images”. También se pueden encontrar por internet varias compañías que comercializan mapas HDR de alta calidad a precios más o menos razonables.

Los apartados que siguen están dedicados a explicar con más detalle la primera posibilidad. La segunda se analiza en la siguiente sección y se basa en algunos programas 3D que, como veremos más adelante, permiten grabar los resultados de un cálculo, en el que ya se manejan valores de un rango dinámico físicamente correcto, en formato HDR; las dificultades, en este caso, como también veremos, son de otro orden. Descarto la tercera posibilidad porque su coste, por ahora, es prohibitivo. Pero no hay que perderla de vista pues esto puede variar en un plazo quizás no demasiado largo. Véase el apartado sobre grabación de panorama con cámaras especiales, más adelante donde se dan algunos datos sobre este tipo de cámaras.

### Grabación de series de fotografías con varias exposiciones

Para generar una imagen HDR a partir de series de fotografías con varias exposiciones se necesita seguir un proceso riguroso para asegurarse de que una misma imagen se ha captado con los diferentes grados de exposición que aseguren que las partes más oscu-

ras (exposición menor) y más claras (exposición mayor) de la imagen resulten claramente visibles. En este apartado se describe este proceso con la advertencia de que habrá que practicarlo varias veces para asimilar adecuadamente todos los pasos.

El proceso descrito es adecuado para escenas estáticas. Hay que evitar que aparezcan personas o animales, que pueden eliminarse pero por métodos de edición trabajosos. También hay que intentar que el encuadre sea exactamente el mismo. Es posible disparar a mano y confiar en un programa de *stitching* para alinear y ensamblar posteriormente las imágenes, tal como se explica más adelante para tomas panorámicas. El resultado será menos preciso pero puede ser aceptable. Sin embargo, hay más garantías de que el resultado sea correcto si las tomas son geométricamente idénticas, por lo que lo más recomendable es utilizar un trípode. Y tampoco está de más utilizar un disparador con retardador o cualquier método que elimine el movimiento o la vibración de la cámara al disparar.

Sería más fácil utilizar *AutoBracketing*, una función incorporada a la mayoría de las cámaras modernas. Pero la mayoría de estas cámaras, incluyendo las de gama alta, tienen un *AutoBracketing* de pocos pasos, de tres tomas de  $+1/-1$  EV, lo que es insuficiente para generar HDR adecuados en la mayoría de los casos. Hay algunas cámaras más recientes que llegan a 5 tomas con intervalos de  $+2/-2$  EV. Esto puede ser suficiente en algún caso pero no en la mayoría. También hay que salir al paso de la idea de que con un archivo RAW de 14 bits se podría obtener un HDR, manipulando el archivo para obtener varios archivos con diferentes exposiciones. El rango dinámico captado no se va a modificar substancialmente por lo que el resultado no daría lugar a un HDR verdadero.

Así que, por ahora (2014), no queda más remedio que recurrir a métodos manuales de variación de la exposición.

Como todo aficionado a la fotografía sabe, hay tres maneras de variar la exposición, va-

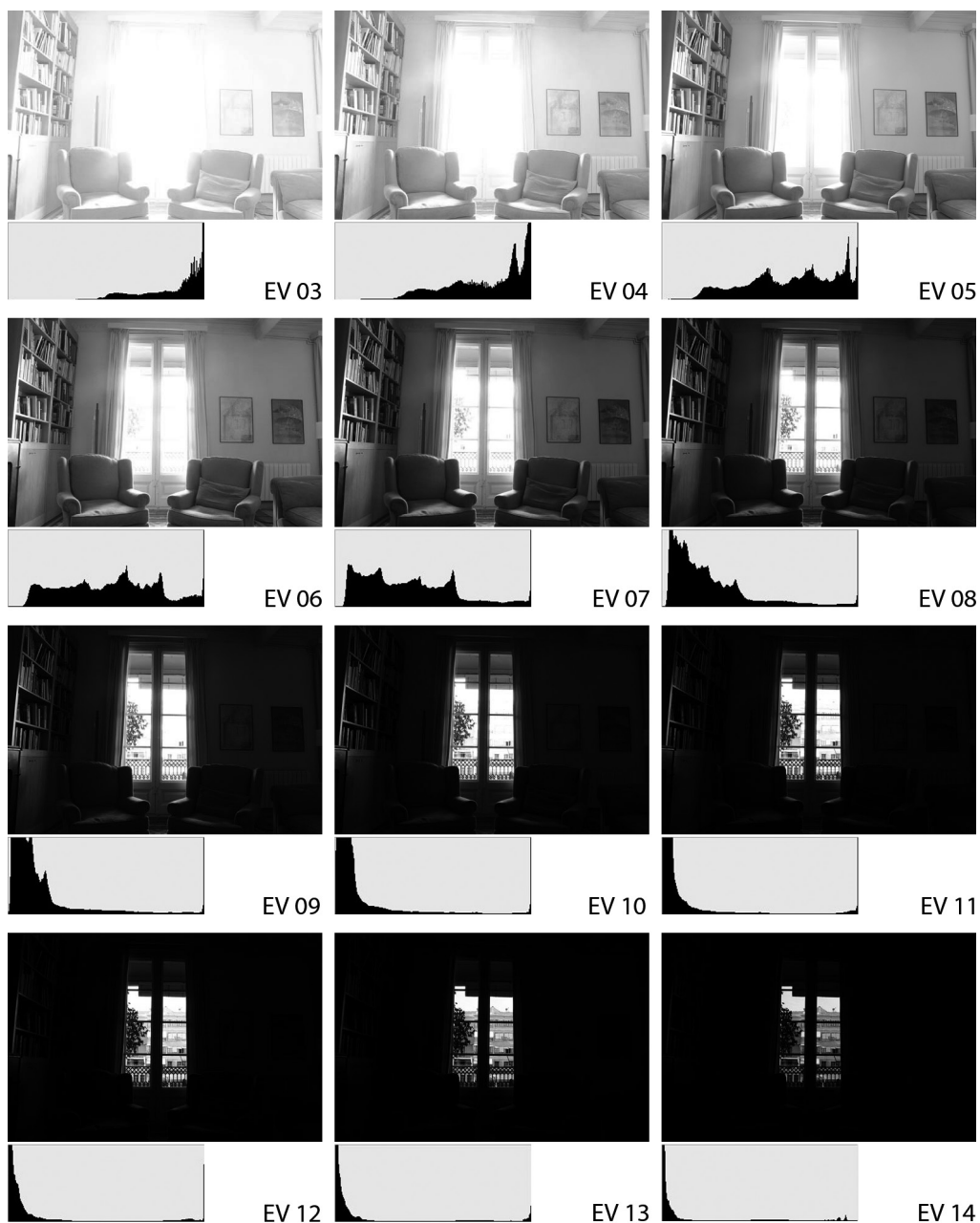


Figura 4.12 Serie de fotografías de un mismo tema con exposición variable entre EV 3 y EV 14, apertura F8 y velocidades variables de 8 segundos (máxima exposición, EV 3) a 1/250 segundos (mínima exposición, EV 14). El histograma resultante se muestra en la parte inferior.



riando la apertura del diafragma, variando la velocidad de la película y variando la velocidad de obturación. Pero solo la tercera es adecuada para obtener imágenes HDR. Si se varía la apertura se puede modificar también la profundidad de campo, lo que impedirá una fusión posterior adecuada. Si se varía la velocidad de la película (ISO) se modifica la calidad del resultado, añadiendo o disminuyendo el ruido, lo que también impedirá una fusión adecuada. Pero si se varía la velocidad del disparo esto no afectará significativamente al resultado, por lo que esta es la opción correcta.

¿Cuántas fotografías hay que tomar y con qué separación de pasos EV? Para captar detalles en las zonas altas, muy iluminadas, el número de exposiciones crece exponencialmente por lo que hay que ponderar muy bien el número adecuado que no puede ser ni muy fiel al detalle, pues resultarían demasiadas tomas, ni demasiado tosco, pues no conseguiríamos lo que queremos. Con las sombras ocurre más bien al revés: puede que nos parezca que los detalles ya quedan expuestos al ser visibles pero la exposición que necesitamos a menudo es mayor. Para poder tomar esta decisión razonablemente, lo mejor es contar con una cámara de calidad media / alta que nos muestre el histograma en pantalla. Si el histograma se desplaza hacia la derecha, hasta que vemos puntos sobreexpuestos, es el momento de detenerse en el valor resultante de velocidad. Luego proceder a la inversa hasta que el histograma se desplace hacia las sombras. Así que la respuesta a esta pregunta sería: en general depende, pero no menos de 5 y tantas como sea necesario para cubrir todo el rango, lo que supondrá, en la mayoría de los casos de 7 a 9. Otro modo de responder a lo mismo es que la mayoría de quienes toman fotografías de este tipo disparan en 1 o 2 pasos de EV. Si se dispara a 1 EV de separación habrá menos ruido y la combinación será más fina; pero el número de fotografías necesarias será el doble. También depende de si se dispara en JPEG o en RAW. En el primer caso será más prudente quedarse en 1 paso EV. En el segundo, con 2 EV será su-

ficiente en escenas con un rango muy amplio. Para la mayoría de los casos se puede disparar en JPEG. Una secuencia que a menudo da buenos resultados es -6, -4, -2, 0, +2, +4, +6. Pero insisto en que depende de los casos y la clave es inspeccionar el histograma de las tomas extremas.

También debe tenerse presente que, en una escena exterior, hay que procurar disparar con rapidez: una nube que pasa puede alterar de un modo importante la exposición entre una toma y otra.

Resumiendo, el procedimiento básico, partiendo de la base de que ya hemos seleccionado el motivo y la hora adecuada para fotografiarlo, es el siguiente:

1. Encuadrar la vista e inmovilizar la cámara. Utilizar un trípode y, a ser posible, un retardador para no mover la cámara al disparar, aunque esto no es imprescindible.
2. Ajustar el balance de blancos según la escena, utilizando preferentemente un modo fijo adecuado al caso. Se puede utilizar el automático pero es más seguro definir un modo concreto.
3. Desactivar el autofocus. Utilizar el enfoque manual y una vez enfocada la escena no volver a tocarlo.
4. Si las tomas se van a hacer en JPEG, asegurarse de que está desactivado cualquier método automático de manipulación del resultado (refinamiento del contraste, del enfoque, etc.).
5. Para que la imagen tenga el menor ruido posible, utilizar valores ISO bajos (no más de 100, o 200 si se utiliza una cámara digital con este límite inferior).
6. Ajustar la apertura a un valor bajo para ganar profundidad de campo:  $f/8$  o  $f/11$  son buenas elecciones en exteriores.
7. Utilizar previamente un modo de exposición con prioridad a la velocidad para anotar el valor resultante de velocidad para este modo. Hacer una estimación de los pasos necesarios y de la velocidad requerida para la exposición más alta y más baja (véase los comentarios dados más arriba).



8. Cambiar a exposición manual, con el valor de apertura que teníamos y el valor resultante de velocidad de obturación inicial para la exposición de referencia.
9. Disparar para obtener la primera imagen.
10. Reducir la velocidad de obturación en 1 o 2 puntos, según los casos (véanse los comentarios dados más arriba). Disparar para obtener la segunda imagen. Seguir reduciendo y disparando hasta el límite que nos hemos fijado al comienzo.
11. Volver a la exposición de referencia. Aumentar la velocidad de obturación en 1 o 2 puntos, como al reducir. Disparar para obtener la segunda imagen. Seguir reduciendo y disparando hasta el límite que nos hemos fijado al comienzo.

Así se obtendría una serie de fotografías como las de la figura 4.13.

#### Montaje y generación de archivos HDR a partir de series de fotografías

Una vez que se cuenta con una secuencia adecuada de fotografías con diferentes exposiciones, la obtención de un único archivo HDR es casi automática si se cuenta con un programa adecuado.

Todo lo que hay que hacer es seleccionar las imágenes y activar el proceso de cálculo del programa escogido. Las imágenes incorporan metadatos en formato EXIF (*exchangeable image file format*, el primer y principal estándar de metadatos para cámaras digitales, creado en Japón en 1998) con una estructura interna basada en el formato TIFF y que incluye entre otras cosas los datos de la toma: apertura, distancia focal, ISO, etc. A partir de estos datos la aplicación puede calcular los datos del nuevo archivo.

Hay varios programas disponibles para esto. La referencia clásica es HDR Shop, desarrollado en la University of Southern California a partir de directrices dadas por Paul Debevec y que era el más utilizado hasta hace unos años. Pero la versión antigua, que sigue

siendo gratuita, ha quedado desfasada y la versión más reciente no es gratuita.

Doy la descripción del proceso a seguir en un par de programas gratuitos y poco conocidos pero muy recomendables y en un programa nada gratuito pero muy conocido (Photoshop). Otra opción intermedia recomendable (no gratuita pero de bajo coste) es Photomatix. Y otra opción obvia sigue siendo HDR Shop. Pueden encontrarse bastantes más por internet.

**Picturenaut** es un programa desarrollado por Marc Mehl y la comunidad de fotógrafos alemanes y del que se hizo una versión en inglés hace pocos años. Es muy utilizado por los fotógrafos pues da resultados de buena calidad. Presenta un interface muy sencillo con opciones para generar HDRs, para *tone mapping*, y para unas pocas cosas más. Es bastante similar a HDR Shop y también puede utilizar plug-in de este programa. Se puede descargar de <http://www.hdrilabs.com/picturenaut/>.

Para generar un HDR a partir de una serie de imágenes con diferentes exposiciones, todo lo que hay que hacer, tras abrir el programa, es lo siguiente.

1. Ir a menú *File/Generate HDR*. Se abrirá un cuadro de diálogo sencillo. Presionar el botón *Add*, desplazarse hasta la carpeta donde están las imágenes y seleccionarlás presionando el botón Ctrl para añadir a la selección o Shift para añadir las comprendidas entre el primero y el segundo clic. Las imágenes seleccionadas aparecerán en una tabla que recogerá sus datos EXIF de apertura y velocidad.
2. Confirmar. Al cabo de unos instantes se generará el archivo HDR.
3. Grabar el resultado. Las opciones para grabar son HDR, EXR, PFM y TIFF. Escoger la primera o la segunda, según lo que admita el programa que vamos a utilizar posteriormente con estas imágenes y los criterios que se han dado en el capítulo 2 y que volveremos a revisar más adelante,



en el capítulo sobre configuraciones de cálculo para IBL.

Esto será suficiente en la mayoría de los casos. Pero hay algunas opciones adicionales que conviene tener presentes. En el primer paso, en el cuadro de diálogo *Generate HDR!* hay más opciones. Además de *exposure correction*, que está activada por defecto, también se puede activar la opción de alineación automática, de balance de blancos y de eliminación de objetos en movimiento (*ghost removal*). Y el procesamiento puede hacerse a partir de una curva lineal, estándar o definida por el usuario (en este caso se activa otro botón para cargar los datos). Para ajustar la visualización del HDR hay disponibles (menú *Image/Tone mapping*) los siguientes algoritmos de *tone mapping*: *bilateral*, *exposure*, *adaptive logarithmic* y *photoreceptor*, cada uno de ellos con sus propios parámetros. Véase la sección sobre operadores de *tone mapping* más adelante, donde se describen estos y otros algoritmos.

Otra opción muy similar, más reciente, es **Luminance HDR** (anteriormente denominado QtPfsGui, por “Qt”, una biblioteca multiplataforma muy utilizada en sistemas abiertos, “pfs”, la biblioteca principal del entorno y la base original del código, y “gui”, *graphical user interface*). Se puede descargar gratuitamente de <http://qtpfsgui.sourceforge.net/>. No describo el procedimiento y las opciones pues son muy similares a las del anterior, aún más simples (quizás excesivamente simples).

**Photoshop** no necesita presentación. Aunque no sea la mejor opción, aquí se describe el método a seguir en este programa que será seguramente la posibilidad más familiar para la mayoría de lectores que no necesiten descargarse más programas. La descripción está referida a la versión CS4 y es prácticamente idéntica a la versión CS6.

1. Abrir la secuencia de archivos que se han captado previamente. Ir al menú *Archivo/Automatizar/Combinar para HDR*. En el

cuadro de diálogo que se abrirá, hacer clic sobre el botón “Añadir archivos abiertos” o “Explorar” si se prefiere seleccionar archivos no abiertos. Dejar marcada la opción “Intentar alinear automáticamente las imágenes de origen”. Confirmar.

2. Al cabo de unos segundos (más o menos, según las características de la imagen y del ordenador) el cómputo concluirá y se abrirá un cuadro de diálogo especial con el resultado en el centro, las imágenes, a un lado, con su exposición relativa (pasos EV) indicada y otros controles que varían según se escoja, en este mismo panel, 16 o 32 bits. La disposición varía en las versiones CS4 y CS6 pero lo fundamental no. Si fuera necesario (por ejemplo, por excesivo desenfoque), desactivar alguna de las imágenes parciales para que no se tengan en cuenta.

A la derecha se mostrará el histograma resultante (si se ha escogido 32 bits) con la opción de modificar la posición del punto blanco o un botón de acceso al mapa de curvas (si se ha escogido 16 bits). Si se desplaza el histograma se verán diferentes resultados pues el monitor solo puede mostrar archivos LDR, no puede mostrar toda la gama de valores. Confirmar para salir del cuadro de diálogo.

3. La imagen resultante puede modificarse cambiando la exposición por medio de un cursor que aparece en la parte inferior (con 32 bits) o entrando directamente en los ajustes de imagen y seleccionando el control de exposición (con 16 bits)
4. Guardar el resultado. Escoger como formato PBM (que puede facilitar el intercambio) o directamente HDR, según lo que se necesite en el programa de simulación que vayamos a utilizar.

## Imágenes panorámicas

Como ya he dicho más arriba, en muchos casos lo que realmente nos interesará será contar con una imagen panorámica HDR. Esto complica el proceso pues tenemos que



utilizar dos tipos de procedimientos y hacer que sean compatibles: un procedimiento para generar el panorama y, por otro lado, un procedimiento para fusionar los valores. Hay varias posibilidades que se discuten en los apartados siguientes.

Pero antes de analizar estas posibilidades es necesario tener presente a dónde vamos a ir a parar, esto es, cuáles son los tipos de panoramas que se utilizan, en general, en simulación visual, con independencia de si son o no HDR.

### Mapas de entorno panorámicos

Necesitaremos, en primer lugar, contar con un mapa adecuado, y “adecuado” quiere decir que envuelva por completo la escena. Por añadidura, si se va a ver como fondo, su proyección debería corresponderse con la que tendría un escenario real desde el punto de vista de la cámara virtual que estamos utilizando. Pero este requisito puede pasarse por alto excepto en casos muy especiales.

Las técnicas que permiten incorporar a una escena virtual este tipo de mapas aparecieron desde los comienzos de la simula-

ción por ordenador. La primera aportación importante a esta línea de trabajo se dio en 1976, cuando Jim Blinn y Martin Newell publicaron un famoso artículo, “Texture and Reflection in Computer Generated Images”, en el que se introducía la técnica de lo que posteriormente ha pasado a denominarse *environment mapping*, si bien se utilizó, en ese caso, para simular reflexiones. Newell utilizó una tetera, la famosa tetera de Utah, sobre la que proyectó una imagen de un interior hecha con un programa de pintura desarrollado por Blinn. El efecto buscado era que el entorno se reflejase en la tetera.

En 1983, Gene Miller y Robert Hoffman presentaron un modelo de iluminación general (publicado al año siguiente) basado en un mapa que rodeaba la escena, asociado a una tabla de valores que se utilizaban como referencia para el cálculo. Un modelo que, en muchos aspectos, se adelantaba a los métodos presentados posteriormente por Debevec para IBL que ya hemos visto. Y en 1986, Ned Greene publicó otro artículo, en que se utilizaba este término, *environment mapping*, como término destacado y se describía un marco general para el uso de esta técnica, con particular énfasis en *cube mapping* que Greene consideraba que era el mejor método para aplicar esta técnica pero que no pudo utilizarse de un modo eficaz hasta 1999, con el lanzamiento de la Nvidia GeForce 256 que incorporó el algoritmo por hardware. En este artículo se incluían métodos para utilizar de modo más eficiente los mapas de entorno, por medio de filtros previos y de otros métodos que corregían problemas de *antialiasing* que afectan particularmente a estas técnicas. Y combinaba una imagen del cielo de 180°, obtenida con un gran angular, con un modelo de un terreno generado digitalmente, para crear un mapa cúbico en cruz, uno de los tipos que veremos más adelante. Estos métodos se incorporaron por primera vez a famosas producciones cinematográficas, por el director de cine James Cameron, principalmente, como *The Abyss* (1989) o

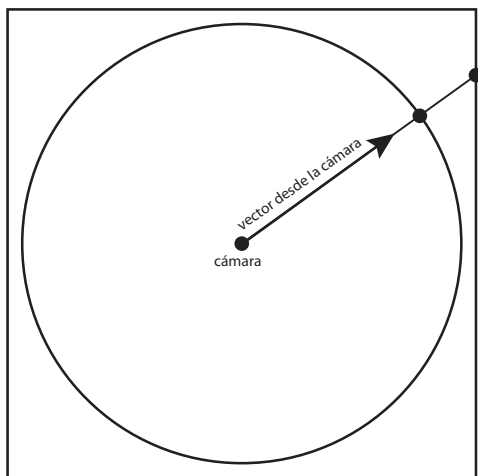


Figura 4.13 Skydomes y skyboxes como mapas de entorno.





*Terminator 2* (1991). George Lucas también las introdujo en ILM (Industrial Light and Magic) para sus producciones cinematográficas.

Si queremos simular la proyección de un entorno sobre un determinado punto de vista, es evidente que la solución intuitivamente más directa es utilizar una esfera que envuelva este punto de vista y proyectar el entorno sobre la esfera. Cada punto de la proyección se corresponderá exactamente con cada punto del entorno representado. Pero dado que las proyecciones que utilizamos son planas, este método tiene el inconveniente de que la proyección sobre la esfera no se puede desplegar sobre un plano sin que aparezcan distorsiones cada vez mayores hasta llegar a una distorsión infinita en los polos. Por otro lado, la correspondencia de proyecciones se mantiene mientras no cambiemos el punto de vista. Pero si lo movemos habrá que rehacer la proyección o aceptar cierto grado de pérdida de correspondencia.

Por estas razones quizás el mejor método es la proyección cúbica: proyectar el entorno no sobre una esfera sino sobre un cubo. Pero las proyecciones esféricas o semiesféricas, también se utilizan, pues pese a los inconvenientes citados también tienen ventajas. La ventaja principal es que se adaptan mejor a escenas en las que solo necesitamos un hemisferio que abarque la escena y que basta con un único mapa para aplicarlo más o menos adecuadamente a este hemisferio. Y, en el caso de IBL, la distribución lumínica es más regular.

Estos dos métodos se denominan corrientemente *skyboxes* (proyecciones cúbicas) y *skydomes* (proyecciones hemiesféricas). Se utilizan principalmente en juegos de vídeo para crear un escenario que se muestre, más allá del escenario principal, desde cualquier punto de vista. Pero también son muy útiles en otro tipo de aplicaciones, principalmente arquitectónicas. En los apartados siguientes se describen técnicas ligadas a estos dos métodos.

## Tipos principales

Supongamos que partimos de un caso ideal, en que hemos cubierto toda la escena que rodea a la posición de la cámara. Es decir, que hemos barrido un ángulo de  $360^\circ$  en horizontal y  $180^\circ$  en vertical, para obtener una serie de imágenes que se adaptarían a una esfera, cuyo centro coincidiría con el punto en que está situada la cámara.

Pero las imágenes digitales son planas. No podemos utilizar una esfera directamente. En cualquier caso, tenemos que desplegar la esfera sobre un plano. Y las posibilidades son tantas como los tipos de proyección que se han utilizado desde el siglo xv para dibujar mapas que representen el globo terráqueo. Y, como es bien sabido, no hay solución perfecta, cada una tiene sus ventajas y sus inconvenientes.

Los programas de *stitching*, que comentaré más adelante, incluyen los principales y algunos más. Por ejemplo, Hugin incluye los siguientes: Rectilinear (el habitual para tomas corrientes), Equirrectangular (el tipo de panorama más utilizado), Cylindrical (el más corriente si se crean panoramas girando la cámara horizontalmente), Mercator (equivalente a cilíndrico pero con menos distorsión), Trans Mercator (igual que Mercator pero girado  $90^\circ$  para adaptarse a objetos altos), Stereographic (una alternativa a *Rectilinear* si se quiere ampliar el campo), Fisheye (equivalente al anterior pero con más distorsión). Y unos cuantos bastante menos habituales: Sinusoidal, Lambert Cylindrical Equal Area; Lambert Equal Area Azimuthal; Albers Equal Area Conic; Miller Cylindrical. Panini; Architectural; Orthographic; Equisolid; Equirrectangular Panini; Biplane; Triplane; Panini General; Thoby Projection. En fin, la mayoría no son sino curiosidades científicas o que sirven para aplicaciones muy especializadas, pero los cito para dejar constancia de la variedad de proyecciones posibles.

Sin embargo, desde el punto de vista de las aplicaciones que nos importan hay tres tipos fundamentales; el primero está incluido en la primera lista que he dado pero los otros dos no. Son los siguientes:



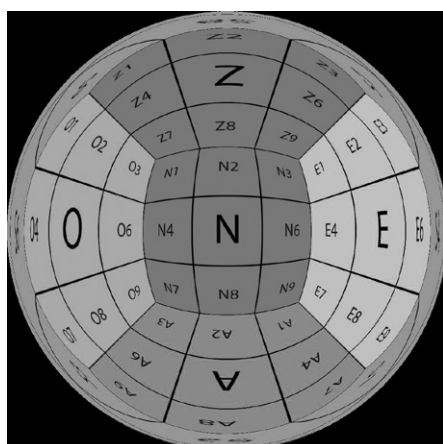
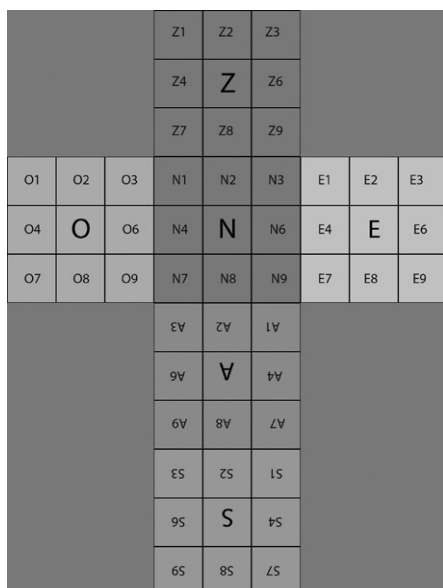
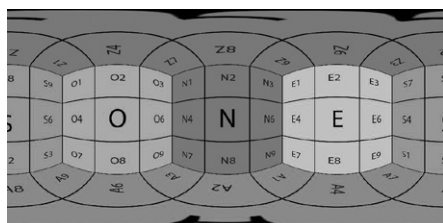


Figura 4.14 Los tres tipos principales de panoramas utilizados en simulación. De arriba abajo: equirrectangular, en cruz vertical, mapa angular. Las letras indican norte, sur, este, oeste (N, S, E, O), cénit (Z) y nadir (A)

**Equirrectangular o latitud/longitud o proyección esférica.** Estas tres denominaciones pueden encontrarse indistintamente según las diferentes aplicaciones. Es la proyección más corriente, que despliega toda la escena sobre un rectángulo de proporción 1:2 que abarca  $180^\circ$  en vertical (latitud) y  $360^\circ$  en horizontal (longitud). Los valores de longitud equivalen a coordenadas x de la imagen y los valores de latitud, a coordenadas y. Al igual que ocurre con todas las proyecciones de una esfera, imposibles de resolver de un modo coherente, hay dos singularidades, dos distorsiones infinitas por la que dos puntos, el cénit y el nadir, los dos polos de la esfera, se transforman en líneas que abarcan todo el rectángulo de proyección. Es importante tenerlo presente pues cualquier modificación de estos puntos dará lugar a errores magnificados. Por otro lado, las líneas verticales de los extremos deben mantener una correspondencia perfecta para que, si son reproyectadas sobre una esfera, no se aprecien “costuras”. Y, por último, como ocurre en todas las proyecciones, las distorsiones afectan de diferente modo a las líneas rectas. En este caso, las líneas verticales se conservan mientras que las horizontales se curvan.

**Cúbica o en cruz horizontal o en cruz vertical.** No es un tipo de proyección canónico. Sin embargo, es el tipo de proyección más utilizado desde hace muchos años en simulación visual, como ya he dicho, pues tiene múltiples ventajas. Si se utiliza como un mapa de entorno permite hacer representaciones con poco coste de computación y buen control de la resolución. La quiebra que cabría esperar en las aristas es imperceptible, no puede distinguirse de lo que se vería con un mapa esférico si el observador está situado en el centro de la escena y los planos están bien creados evitando costuras entre ellos. Por estas razones, es el método utilizado corrientemente en juegos de vídeo o, por ejemplo, en el VR de Quick Time u otras aplicaciones similares. Y muchas tarjetas gráficas tienen incluso zonas de memoria dedicadas a procesar este tipo de mapas cú-

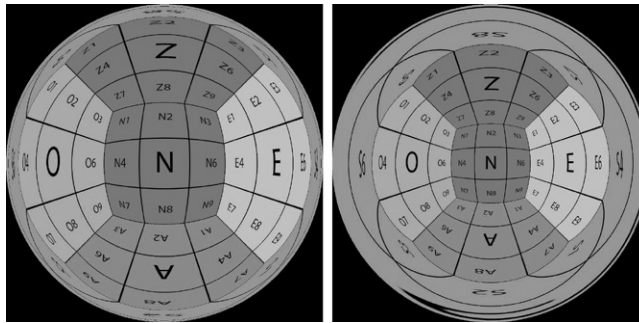


Figura 4.15 Diferencias entre tipos de proyección usados en HDR Shop: izquierda: mapa angular; derecha: mirrored ball.

bicos. Para generar este mapa, como veremos, hay que tomar 6 vistas con un ángulo de  $90^\circ$  (o algo más para contar con un solape que nos permita fusionarlas fácilmente). Estas vistas se pueden utilizar de modo independiente o componer de diferentes modos, lo que depende generalmente de los requisitos del programa en que se vayan a utilizar. La composición se puede hacer, como también veremos, por medios automáticos o manuales. El modo más corriente es en cruz horizontal (*Horizontal Cross*) pues facilita la edición, aunque también es frecuente el modo en cruz vertical (*Vertical Cross*). Otra de las ventajas importantes de este método es que no hay distorsión, las líneas rectas verticales y horizontales se conservan. Pero una desventaja también importante es que nos obliga a cargar en memoria más del doble de lo que necesitamos pues, en general, solo nos interesará visualizar una parte de este mapa. Esto puede evitarse, en la práctica, descomponiendo el mapa en las partes que nos interesen si solo se va a utilizar como fondo.

**Mapa angular o bola especular** (*angular map, mirrored ball, light probe*). Este tipo de proyección no se encontrará en los tratados o páginas web sobre tipos de proyecciones pues puede decirse que su nacimiento está ligado a las técnicas de IBL introducidas por Paul Debevec hacia 1998 y al uso de bolas especulares para captar la iluminación de

una escena. En HDR Shop se encuentran 3 tipos similares: *mirrored ball*, que abarca  $360^\circ \times 360^\circ$ , *mirrored ball close up*, que abarca solo los  $180^\circ$  centrales, *angular map* o *light probe*, que es similar a la *mirrored ball* con la diferencia de que la dimensión radial se proyecta linealmente en función del ángulo, lo que hace que los bordes queden mejor distribuidos. Para evitar confusiones, en la figuras adjuntas, 4.15 y 4.16, me refiero solo a bola especular (*mirrored ball*) y mapa angular (*angular map/light probe*).

La proyección clásica más cercana a este tipo es la estereográfica (*stereographic*) o la proyección en ojo de pez (*fish-eye*). Sin embargo en estas, el campo abarcado es de  $180^\circ$ . Pero en la proyección angular es, idealmente, de  $360^\circ$  y, prácticamente, de un valor bastante cercano, algo que puede resultar sorprendente pero que, como veremos más adelante, es el campo que corresponde a este tipo de proyección. También veremos que cuanto más pequeña sea la bola y más lejos esté la cámara, más nos acercaremos a la situación ideal en que los  $360^\circ$  quedan cubiertos. En la práctica, obviamente, esto no es posible y hay una zona que queda oculta por la propia bola. Y, lo que es más importante, hay una zona de distorsión importante que coincide con la línea de tangencia. Por otro lado esta proyección tiene la ventaja de que es continua, solo se rompe en un punto singular. Y la desventaja de que no se presta a ser editada si no se despliega.



### Software de generación de panoramas

En general, nos encontraremos con imágenes corrientes que habrá que combinar de diferentes modos. De acuerdo con las finalidades de este capítulo podemos clasificar estos programas en tres grupos:

- Programas sencillos que facilitan la agrupación y desagrupación de mapas cúbicos. De este tipo son programas como `cube2cross.exe` o `cube the sphere`, que se pueden descargar gratuitamente desde internet.
- Programas especializados en las tres proyecciones características que hemos visto más arriba: Equirrectangular, Angular y Cúbica. De este tipo es el programa HDR Shop.
- Programas de *stitching*, que permiten generar panoramas equirrectangulares a partir de varias imágenes, detectando puntos homólogos y fusionándolos automáticamente.

Los dos primeros los veremos más adelante a través de ejemplos específicos. En este apartado haré una introducción general a los terceros que también se utilizarán más adelante con otros ejemplos.

La creación de imágenes panorámicas se relaciona con otros temas que merece la pena señalar de pasada. Crear panoramas puede ser una finalidad en sí misma, lo que entronca con una tradición de pintura clásica o puede ser una finalidad supeditada a otras, como la necesidad de visualizar un escenario complejo o de crear fondos para escenas. O bien, como en nuestro caso, puede ser una finalidad ligada a crear imágenes que sirvan para iluminar escenas.

Los panoramas modernos se crean con técnicas digitales. Más exactamente, con una técnica ya mencionada denominada *stitching* (sutura, costura). Esta técnica es imprescindible para montar panoramas digitales pero también puede utilizarse con otras finalidades, entre ellas las de crear texturas “sin cos-

tura” (*seamless*). En este apartado me limitaré a describir todo lo que esté relacionado con la finalidad principal a que apuntan estas técnicas: crear mapas de entorno que puedan ser utilizados, principalmente, en sistemas de iluminación IBL.

Hay varios programas de *stitching* que permiten crear panoramas con facilidad a partir de series de imágenes. Menciono solamente los dos que me parecen más recomendables, Hugin y PTGUI. También puede utilizarse el Photomerge de Photoshop. Pero las prestaciones son menores.

**Hugin** es un programa gratuito que se desarrolló a partir de 2007 aproximadamente y que consiste básicamente en un GUI (*graphic user interface*) que facilita el uso de una serie de algoritmos, agrupados bajo la denominación *panorama tools*, desarrollados principalmente por Helmut Dersch, un profesor de física y matemáticas alemán, a partir de 1998. Dersch tuvo que detener el desarrollo de estas herramientas en 2001 debido a problemas legales con la compañía IPIX (una compañía estadounidense, con base en Virginia y que trabajaba en este campo). A partir de esa fecha, Dersch liberó la biblioteca básica (`pano12`) y varios programas y algoritmos agrupándolos en una GNU general public license. El término *GNU* es un acrónimo recursivo que significa “GNU is Not Unix” y es la denominación de un sistema operativo creado como alternativa a Unix a partir de 1983 por Richard Stallman, un programador de Nueva York, fundador del movimiento mundial por un software libre.

El desarrollo del código de los *panorama tools* se continuó por varios colaboradores externos y en 2003 se inició un proyecto que fue acogido por SourceForge (un repositorio internacional de proyectos desarrollados en código abierto). En 2007 Dersch cambió la licencia por una *GNU lesser general public license* (más permisiva que la anterior). Véase [http://wiki.panotools.org/Panorama\\_tools](http://wiki.panotools.org/Panorama_tools) para más información sobre Pano Tools.

Hugin puede descargarse de <http://sourceforge.net/projects/hugin/>



**PTGUI** se desarrolló originalmente también como un GUI para *panorama tools*, como se refleja en su propio nombre (las siglas corresponden a *panorama tools graphical user interface*). Pero posteriormente evolucionó para convertirse en un programa comercial (la versión corriente costaba unos 80 \$ y la Pro, 150 \$ en 2012) muy utilizado por fotógrafos profesionales. Su aspecto y funcionalidad son muy similares a las de Hugin aunque el funcionamiento es más fluido y más fácil de asimilar.

Ambos programas se basan, como otros, en una serie de procedimientos básicos que resumiré a continuación, partiendo de la base de que ya contamos con una serie de fotografías obtenidas tal como se ha descrito en apartado Grabación de series de fotografías solapadas. Convendrá revisar este apartado para recordar las características que deben tener las tomas y doy por supuesto que tras esta primera fase contamos con una serie de fotografías adecuadas, con similar exposición y características y un buen porcentaje de solapamiento.

Comenzaré por describir la técnica general de *stitching* para imágenes corrientes, en LDR, grabadas en un formato habitual como JPEG, PNG o TIFF. El resumen que sigue es muy sintético y si no resulta suficiente habrá que completarlo con la ayuda del programa. El proceso elemental consta, tanto en Hugin como en PTGUI (cada paso se corresponde con un comando fácilmente localizable) de los pasos siguientes:

1. Carga de las imágenes. Al cargar las imágenes el programa reconoce automáticamente sus características principales (apertura, exposición, etc.) que están grabadas como datos EXIF.
2. Alineación de las imágenes. Por medio de una serie de algoritmos incorporados al programa, se detectan puntos de control homólogos en las diferentes imágenes. Los puntos de control generados se pueden editar y ajustar. También pueden crearse puntos de control manualmente, lo que para aplicaciones especiales puede ser necesario pero en nuestro caso, con

un poco de suerte, no. Cuanto mayor sea el solapamiento de las imágenes más fácil será que el programa detecte automáticamente puntos de control.

3. Optimización de la correlación. Este paso puede no ser necesario en la mayoría de los casos. A partir de los puntos de control encontrados y de una configuración de optimización seleccionada por el usuario, se lleva a cabo una correlación virtual entre los puntos de control. Si el resultado es aceptable, se puede continuar y, si no, cabe la opción de reajustar los puntos de control.
4. Optimizar la exposición. Como el anterior, este paso tampoco será necesario, excepto en algunos casos. También pueden optimizarse los ajustes de exposición que son necesarios para que la fusión entre las imágenes elimine las diferencias de color e intensidad entre las imágenes. Como en el caso anterior esta fase admite cierto grado de interacción con el usuario.
5. Previsualización del panorama y elección y ajuste del tipo de proyección. El resultado se reproyecta según la finalidad del panorama (según el tipo escogido de los que se han descrito más arriba), y en cada caso hay controles adicionales para ajustar el ángulo de visión (FOV, *field of view*) en horizontal y vertical así como otros ajustes que no son relevantes en nuestro caso. En la mayoría de los casos, la proyección que nos interesará es la equirectangular.
6. *Stitching*. Al activar este último comando el proceso definitivo se pone en marcha y las imágenes se fusionan con las características y el tipo de proyección especificados en los pasos anteriores. El resultado puede guardarse en los formatos habituales LDR (JPEG, PNG, TIFF...) y, además, en HDR (HDR, EXR).

### Transformaciones de panoramas

Los programas mencionados permiten hacer transformaciones de uno a otro tipo



de panorama. En PTGUI o Hugin todo lo que hay que hacer, en principio, es escoger el tipo correspondiente desde el panel de panoramas y la transformación se hará automáticamente (aunque la primera vez, probablemente los resultados no serán los esperados). Sin embargo, para las finalidades que nos interesan, será mejor utilizar HDR Shop (basta con la primera versión que sigue estando disponible de modo gratuito). En el ejemplo que sigue se muestra cómo pasar de una a otra de alguna de las proyecciones básicas que obtendremos en las secciones siguientes.

Supongamos, para empezar, que ya contamos con una imagen montada en cruz vertical con alguno de los procedimientos que veremos más adelante. Para convertirla a un panorama equirectangular (*latitude/longitude*) o angular (*angular map/light probe*) hacer lo siguiente. Partimos la base de que ya contamos con HDR Shop (lo que sigue está basado en la versión 1.0.3, de 2001).

1. Desde HDR Shop, abrir la imagen en cruz que se supone que ya hemos generado (menú *File/Open*). Utilizar las herramientas incluidas en menú *View* para visualizarla adecuadamente.

2. Activar las transformaciones de panorama. Para ello, ir al menú *Image/Panorama/Panoramic Transformation*. Se abrirá un panel, con dos grupos, uno a la izquierda y otro a la derecha.
3. En el grupo de la izquierda, *Source Image*, comprobar que está seleccionada nuestra imagen y que el formato es Cubic Environment (Vertical Cross). Dejar el resto de valores tal como están. Desmarcar la opción *interpolation* (bicubic por defecto) pues puede dar lugar a líneas negras que aparecen en el mapa final.
4. En el grupo de la derecha, *Destination Image*, dejar la opción predeterminada para el destino, New Image para que se cree una nueva imagen. En *Format*, desplegar la lista colgante y escoger una de las opciones que nos interesan: en este caso, *Latitude/Longitude*. Mantener activada la opción Use suggested Aspect Ratio (2:1). Dar un valor de salida adecuado para *Width* y *Height*, por ejemplo, 2.048 x 1.024. Confirmar, para que se genere la imagen. Cuando se haya concluido el proceso guardarla con un nombre adecuado y en formato HDR. Repetir la misma operación, y escoger esta vez la otra opción: *Mirrored Ball*. Mantener activada la opción Use sugges-

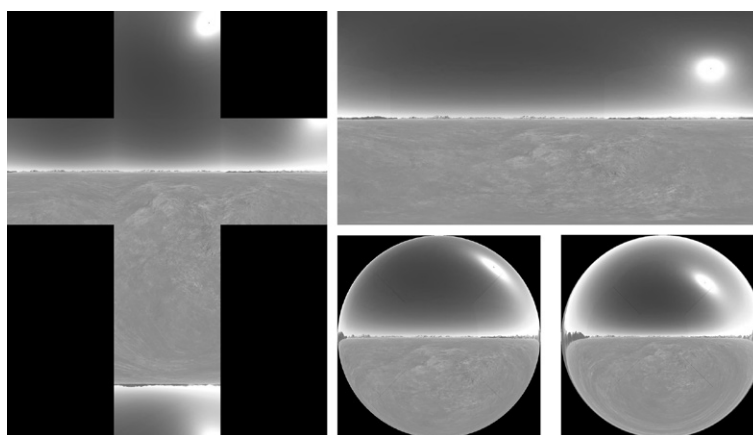


Figura 4.16 Resultados de la conversión del mapa cúbico vertical en cruz (izquierda): arriba a la derecha, equirectangular; abajo a la derecha, bola especular (izquierda) y mapa angular (derecha).





ted Aspect Ratio aunque no coincidirá con el formato de salida que es 1:1. Dar un valor de salida para *Width* y *Height*, por ejemplo 1.024 x 1.024. Confirmar para que se genere la imagen. Cuando se haya concluido el proceso guardarla con un nombre adecuado y en formato HDR.

Repetir la misma operación, y escoger esta vez la opción: *Light Probe*. El resultado es muy similar al anterior pero el ángulo que se abarca es algo mayor.

Tanto la *mirrored ball* como el *angular map* cubren aproximadamente 360°, como ya he mencionado. El mapa angular comienza a inclinar el entorno a mayor distancia del borde de la esfera, como puede apreciarse en la figura 4.17. Por esta razón, distribuye mejor los valores, comprimiéndolos algo menos en los bordes. Debido a esto, los mapas angulares (o los esféricos, que son similares) son más adecuados para utilizarlos como mapas de fondos en simulación visual.

Si vamos a utilizar estos mapas como fuente de iluminación, no es demasiado importante que representen muy bien el fondo, como ocurre con mapas panorámicos que se utilizan para simular un fondo. De hecho, puede interesar que sean más bien difusos y con poco detalle para que no den lugar a frecuencias altas que se traducirían en defectos en la distribución de la iluminación, lo que a su vez se traduciría en defectos en la imagen final. Como veremos más adelante, a menudo se utilizan dos mapas similares, uno como fuente de iluminación y otro como fondo. El primero puede ser un mapa angular mientras que el segundo puede ser un panorama equirectangular recortado (la parte inferior no es visible en muchos casos). También se puede utilizar el mapa angular como una fuente de iluminación no visible, representar el fondo como un canal alfa y componerlo después con un programa de tratamiento de imágenes.

### Skydomes y skyboxes

Como ya he dicho antes, los dos tipos más corrientes de utilización de panoramas son los que se denominan cúpula celeste o *skydome*

y caja celeste o *skybox*. Usaré los términos ingleses pues las traducciones que he dado no son muy frecuentes.

Una *skydome* consiste en un mapa que se adapta a una esfera o un hemisferio. En teoría tiene dos ventajas importantes, que también he mencionado anteriormente. La primera, que solo utiliza un mapa, por lo que su preparación y aplicación es más sencilla aunque, como veremos, paradójicamente, uno de los métodos más efectivos de crear este tipo de mapa es creando primero un mapa cúbico, adecuado para una *Skybox* y, de ahí, obtener por transformación un mapa esférico o asimilable a un mapa esférico. La segunda ventaja es más teórica y deriva de que, en principio, la relación de los puntos del mapa con el espectador es más coherente. Pero es una ventaja inestable pues solo serviría para el caso en que este punto de vista se mantuviera fijo. Pero las desventajas son importantes y también se han mencionado, principalmente las distorsiones, que se intensifican en los polos. La segunda sería la dependencia del punto

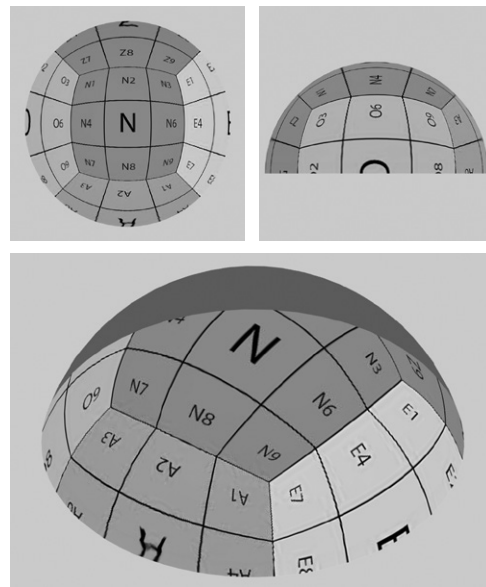


Figura 4.17 Skydome con un panorama equirectangular aplicado: a) vista inferior, b) lateral, c) perspectiva.



de vista. Y la tercera, la menor eficiencia de procesamiento. Por estas razones, a menudo se prefiere la utilización de *skyboxes*.

En la práctica se aplica a un hemisferio y por lo general se utilizan panoramas equirrectangulares (de proporción 2:1) de los que solo aparecerá la mitad superior, tal como se puede comprobar en la figura 4.18. Un panorama estereográfico puede ser preferible pues hay menos compresión en los polos. Más adelante analizaremos los procedimientos específicos y las ventajas e inconvenientes de cada uno.

Como decía, un recurso habitual, un tanto paradójico, es crear un mapa cúbico, adecuado para una *skybox* y, luego, convertirlo a un

panorama, como veremos también más adelante. No es una solución muy satisfactoria pues el horizonte queda en medio y lo que interesa es abarcar la mayor parte posible de cielo. Pero este problema es difícil de solucionar en cualquier caso, pues si se rotase la cámara para ver más cielo las vistas no encajarían lateralmente.

Una *skybox* consiste en un mapa desplegado sobre las seis caras de un cubo virtual y proyectado mediante una técnica conocida como *cube mapping*, una variante de *environment mapping*. Aunque las caras son planas y se encuentran en aristas y es-

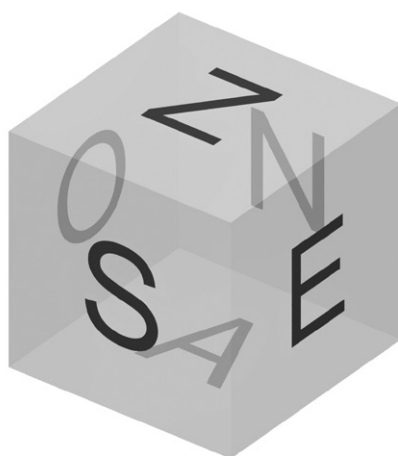


Figura 4.18 Skybox. Orden de vistas del cubo envolvente.

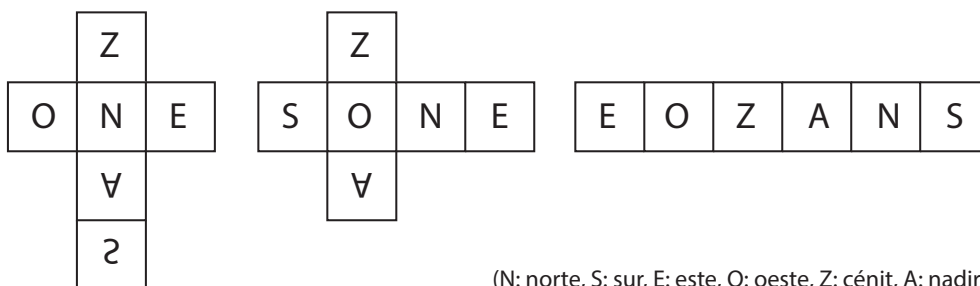


Figura 4.19 Skybox. Tipos de despliegue: a) en cruz vertical, b) en cruz horizontal, c) horizontal (formato Nvidia DDS Exporter).





quinas, un observador situado en el medio recibiría la ilusión de un espacio continuo 3D. Al moverse, las proyecciones permanecen en su sitio, lo que da por añadidura la impresión de que están muy lejos pues otros objetos cambian su posición relativa con respecto a este fondo. Por esta razón, conviene que los mapas representen paisajes lejanos o edificios situados también muy lejos de la escena.

Para generar una *skybox* se hace una representación de la escena seis veces, apuntando en cada una de las direcciones de un cubo envolvente, con un ángulo de visión de 90° y una resolución de salida cuadrada y preferentemente potencia de 2. El resultado es un mapa cúbico, un conjunto de seis imágenes cuadradas que se proyectan sobre un cubo para crear un panorama de 360°. Las tarjetas gráficas cuentan con recursos especiales para gestionar estos mapas, por lo que su uso tiene la ventaja de que se representan con gran rapidez.

Por lo general, se usan de tal modo que la cámara está situada en el centro del cubo con lo que las imágenes parecen situadas en el infinito. El centro de cada cara está más cerca que las aristas por lo que hay cierto grado de distorsión, más notoria si se contempla el cubo desde otro ángulo. Para contrarrestar este efecto en algunos casos se introduce cierto grado de distorsión angular.

El orden en que se generan las imágenes debe quedar claramente determinado para facilitar el montaje posterior. Hay algunos criterios que varían según las aplicaciones. Si se va a utilizar de modo semimanual, como será nuestro caso, el orden es irrelevante con tal de que funcione. Las figuras 4.19 y 4.20 muestran algunos ejemplos característicos de denominación y ordenación.

Si se utiliza como un mapa unificado se pueden desplegar en tres tipos de formatos característicos: cruz horizontal, cruz vertical y horizontal, sin cruz. Este último es el formato utilizado por Direct3D para el formato DDS. La figura 4.20 muestra un esquema de estos tres tipos.

Para obtener estos mapas unificados hay dos alternativas principales: montar las imágenes manualmente en un programa de edición de imágenes como Photoshop o utilizar algún programa que lo haga de modo automático, como se explica más adelante.

Si los ángulos horizontal y vertical de la cámara abarcan exactamente 90° las seis imágenes se podrían componer directamente. Pero debido a pequeñas imperfecciones podrían aparecer líneas en las uniones, más aún con un sistema de cálculo avanzado de iluminación. Para evitar este problema en algunos casos se prefiere usar un campo mayor, por ejemplo 110°, para que los bordes se solapen y luego fusionar y corregir manual o automáticamente las suturas.

## Métodos de grabación y montaje de panoramas HDR a partir de escenarios reales

### Grabación de series de fotografías solapadas

Para crear panoramas, la primera alternativa es partir de varias tomas que luego se podrán combinar con un programa de *stitching*.

Esto plantea varias cuestiones técnicas. La primera es cuáles son los requisitos de las fotografías para poder ser fusionadas adecuadamente en una única imagen con los programas citados. Esta cuestión es independiente de si las imágenes son HDR o no. La segunda es si el procedimiento debe llevarse a cabo fusionando primero las imágenes de igual exposición para obtener varios panoramas con diferente exposición y, luego, fusionar estos panoramas obteniendo un panorama HDR o bien, si es preferible crear archivos HDR a partir de series de tomas fijas y luego fusionar estos archivos HDR en un panorama.

Comencemos por el primer problema. Para crear un panorama a partir de varias imágenes había un método relativamente tradicional que no hay que descartar. Para que el proceso de fusión se pudiera hacer con facilidad, cuando los recursos digitales eran más limitados o se requería una alta precisión, la cámara se mon-



taba sobre una rótula especial que había que incorporar al trípode. Una alternativa menos precisa sería localizar el punto nodal de la cámara y procurar (por diversos medios que pueden ir desde el puro sentido del equilibrio a colgar una plomada desde ese punto o apoyarlo en algún lugar fijo) que la cámara gire en torno a dicho punto para minimizar los desajustes. Hay recursos caseros que se pueden encontrar por internet buscando por las palabras clave “camera rotation nodal point”.

Y una alternativa aún menos precisa es procurar que la cámara gire manteniéndose vertical y sin desplazarse mucho: si no hay objetos que importen en primer plano puede obviarse este problema cuidando tan solo la cámara con cuidado. Los recursos digitales han simplificado el proceso pues pueden fusionarse varias imágenes aunque se hayan obtenido desde posiciones diferentes. Este proceso se denomina *stitching* (del inglés *to stitch*, coser o suturar) y hay varios programas que ya he citado de software de generación de panoramas que lo hacen muy bien. El proceso se basa en identificar, automática o manualmente, una serie de puntos homólogos en pares de fotografías.

Sin embargo, conviene tener en cuenta que, para que este proceso resulte fiable, hay que tener en cuenta varias cosas:

Las fotografías deben tomarse con la misma configuración: apertura, velocidad, balance de blancos y enfoque. Por esto conviene tomar medidas de exposición previas y luego escoger el modo manual, con los valores que sean. También, por estas razones, puede ser preferible tomar las fotos en modo *Raw*, lo que permitirá ajustar estos valores *a posteriori* aunque en principio no es necesario si se toma la precaución de ajustar bien la configuración de entrada. Si hay variaciones importantes de iluminación, tomar los valores de exposición apuntando a una zona intermedia. Recapitulando todo lo dicho:

- Las fotografías deben solaparse al menos en un 20 % o un 25 % (algunos prefieren hasta un solape de 50 % en determinados casos).

- Deben evitarse objetos situados en primer plano que den lugar a errores de paralaje si no se quiere utilizar mecanismos especiales para controlar el giro de la cámara, tal como se explica en el apartado siguiente.
- El procedimiento descrito antes (giro a partir de una rótula o procurando rotar la cámara sobre su eje principal, etc.) facilita considerablemente la fusión por lo que no hay que descartarlo si se puede contar con un aparato de este tipo.

Hay que tener en cuenta que para cubrir por completo una escena habría que tomar más de 30 fotos: unas 4 filas y unas 8 columnas. Pero en muchos casos bastará con la mitad y en otros, con menos. Si se requiere una panorámica completa habrá que terminar con una foto a 90° apuntando al cenit y otra a -90 apuntando al nadir. Habrá un puntito negro donde se junten que habrá que eliminar por métodos manuales.

Algunos profesionales calculan el número de tomas mediante la fórmula aproximada  $n = 1,3 \times (hFOV_{delPanorama} / hFOV_{deLaLente})$  en donde 1,3 es un factor relacionado con el solapamiento y hFOV es el ángulo de campo horizontal (*horizontal field of view*). Así, por ejemplo, si nuestro objetivo abarca 55° y queremos abarcar 360°, necesitaremos entre 8 y 9 tomas ( $1,3 \times 360/55 = 8,5$ ). Y si nuestro objetivo abarca 90° necesitaremos unas 5 o 6 tomas ( $1,3 \times 360/90 = 5,2$ ).

Un número tan elevado de tomas multiplica los problemas técnicos que comento más abajo. Para evitar tal elevado número de tomas merecerá la pena considerar la posibilidad de adquirir un objetivo en ojo de pez de bajo coste, tal como un Samyang de 8 mm. Un objetivo de este tipo no proporciona una imagen circular que abarque 180° a no ser que se utilice con una cámara *full-frame*, lo que dispara aún más el coste. Pero si se utiliza con una cámara SLR normal proporciona 180° en diagonal, que equivalen aproximadamente a 150° en horizontal y 110° en vertical. Si giramos la cámara para invertir estos valores tendremos 150° en vertical y 110° en ho-

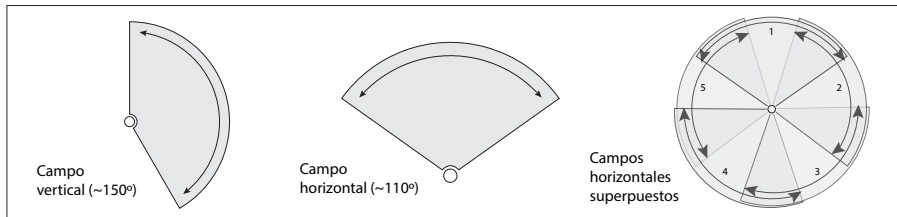


Figura 4.20 Panoramas. Grabación de series solapadas: a) tomas individuales, b) esquema de las tomas, c) composición resultante (2,5:1), d) Composición corregida a 2:1.



horizontal. Esto quiere decir que nos bastarían 4 tomas para barrer toda la escena en horizontal 360°, aunque el solape será pequeño y deberemos extremar la precisión y, por tanto, será mejor, si no contamos con una rótula o un dispositivo para girar sobre el punto nodal, 5 tomas, como en el ejemplo que sigue. Y si giramos la cámara 20° hacia arriba para que el cénit entre en la imagen podemos obviar la parte inferior que, por lo general, no interesa a la hora de generar panoramas pues quedará oculta por el suelo de la escena virtual.

Las imágenes de la figura 4.21 ilustran estos casos. Si utilizáramos una cámara relativamente normal, con un objetivo de, por ejemplo, 18/105 mm, que también es relativamente corriente, tendríamos que obtener al menos 7 tomas en horizontal y 2 en vertical, es decir, un total de 14 tomas. Si utilizamos una cámara con un objetivo de 8 mm podemos reducir esta cifra a 5 tomas totales.

El segundo problema es decidir el orden del proceso. Pero esto lo discutiremos al final de esta sección, en el apartado que incluye una descripción general previa de los programas de *stitching*, Montaje y generación de HDR panorámicos.

Parto por tanto, de la base de que, quien quiera obtener panoramas HDR contará al menos con una buena cámara SLR, un objetivo de 8 mm y un trípode. A partir de aquí, en lo que sigue parto también de la base de que serán suficientes 5 tomas, cada una de ellas con 7 exposiciones separadas por 1,5 EV. Según los casos puede ser preferible aumentar este valor a 2 EV o reducirlo a 1 EV. Pero el valor adoptado parece suficiente para cubrir el rango dinámico en la mayoría de los casos. Como he explicado antes, la apertura se mantiene fija y se varía la velocidad. Para acelerar las tomas convendrá utilizar *bracketing*, algo que está incorporado con al menos 3 pasos en la mayoría de las cámaras modernas. Con todos estos supuestos, el procedimiento general sería el que sigue.

1. Lo primero es buscar un lugar y un día y una hora adecuados. El lugar debe ser

una explanada abierta en todas direcciones y donde el cielo, que es lo que nos interesa, domine el panorama. El día debe ser preferentemente algo nublado, pues el resultado será más interesante y también evitaremos que el Sol sea visible, lo que complica el proceso. Y sin viento, para que las nubes no cambien de posición entre tomas. La hora deberá ser la primera de la mañana o última de la tarde para que el sol esté bajo y salga en una sola de las imágenes. Las imágenes de la figura 4.21 están tomadas en junio a las 20:00.

2. Antes de desplazarse al lugar:

- Si el trípode no tiene un buen sistema de anotaciones de ángulo, marcar con un lápiz en su cabezal los puntos de giro. Un recurso casero pero eficaz es tomar un trozo de cuerda, enrollarla alrededor del trípode y dibujar unas marcas al principio y al final de toda la vuelta. Luego dividir este segmento en 5 partes iguales y, manteniendo la cuerda enrollada, llevar estas marcas al trípode con un lápiz. Así controlaremos con facilidad el giro horizontal.
- Situar la cámara con el trípode debajo de algún elemento del techo que caiga sobre la vertical trazada desde la entrada de la lente. Girar la cámara en vertical hasta que este elemento coincida con el borde de la imagen. Así, tendremos suficientemente controlado el giro vertical. Se puede llevar la cámara ya montada con este ángulo o hacer también una marca en el cabezal del trípode.

3. En el lugar:

- Comprobar, en el menú de configuración de la cámara, la resolución y la calidad (resolución máxima y JPEG en calidad máxima es la mejor opción).
- Ajustar el balance de blanco a sol o nublado, según sea el caso.
- Comprobar la imagen inicial, que el protector esté bien ajustado, que no aparezcan elementos indeseados en los márgenes, etc.



- Ajustar la exposición. Las fotos de la figura 4.21 están basadas en una exposición media con  $f/11$  de apertura y  $1/180$  de velocidad pues la hora era bastante tardía. En general la velocidad será seguramente superior a  $1/500$  para esa apertura. En cualquier caso, a partir de aquí, reducir y aumentar la velocidad hasta comprobar que el histograma se desplaza hacia los extremos.
- Ajustar el *bracketing* a 3 frames con 1,5 de separación.
- Enfocar. Comprobar que no hay sombras o interferencias en la imagen.
- Disparar. Girar el dial hacia un lado seis pasos y disparar tres veces en sucesión, luego hacia el otro lado otros seis pasos y volver a disparar tres veces.
- Girar hacia la derecha hasta encontrar la primera marca del trípode y repetir los disparos. Seguir girando hasta completar las cinco tomas. Disparar con rapidez (y sin necesidad de mirar, excepto en la primera) para evitar cambios de luz.

Así se obtendrán 5 series de fotografías con 7 valores de exposición. Las velocidades de obturación de las fotografías adjuntas, para el valor de apertura mencionado ( $f/11$ ), fueron las siguientes (entre paréntesis el EV equivalente):

- 1) **1/8** (ev10), 2) **1/20** (ev11,5), 3) **1/60** (ev13),  
4) **1/180** (ev14,5), 5) **1/500** (ev16), 6) **1/1.500**  
(ev17,5), 7) **1/4.000** (ev19)

es decir, 7 pasos EV separados por 1,5. Como he dicho antes, en otros casos esto puede ser insuficiente y será preferible 7 pasos EV separados por 2,0 (de EV 6,0 a EV 18,0, por ejemplo).

Convendrá revisar las fotografías obtenidas y reordenarlas pues se pueden haber producido duplicaciones.

Las imágenes de la figura 4.21 muestran, a) Las 5 tomas, b) Un esquema de los campos visuales abarcados en vertical y en horizontal y de la superposición en horizontal de las 5 tomas, c) El resultado final del proceso,

d) El resultado final del proceso corregido para obtener un panorama equirrectangular de proporción 2:1.

El proceso seguido para obtener estas dos últimas figuras se resume más adelante, en el apartado Montaje y generación de HDR panorámicos.

### Grabación por medio de una bola especular (*mirror ball*)

La segunda alternativa, también muy utilizada profesionalmente, es tomar una fotografía de



Figura 4.21 Grabación con una bola especular. Bola (sobre una almohadilla gris) utilizada para generar las imágenes panorámicas que siguen.

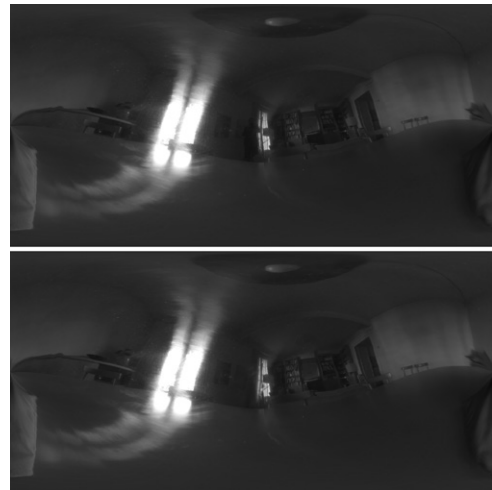


Figura 4.22 Grabación con una bola especular: a) resultados originales, b) resultados recortados y manipulados para eliminar el fotógrafo.

una bola especular que refleje todo el entorno. Este recurso se utiliza no solo para crear panoramas sino también para captar la iluminación. Me referiré a este segundo uso en la segunda parte de este apartado.

Prescindiendo de la discusión sobre el tipo de proyección obtenido y de las aplicaciones correspondientes, que se verán más adelante, en este apartado me limitaré a resumir los aspectos prácticos. Es decir, qué se necesita para obtener una imagen de estas características y cuáles son los pasos a seguir.

El equipamiento necesario es: a) una buena cámara digital SLR, b) un trípode, c) una bola especular, d) opcionalmente, un trípode secundario para la bola.

Doy por supuesto que quien lea esto cuenta con las dos primeras cosas, la cámara y el trípode o sabe cómo conseguirlas. La tercera no parece difícil de conseguir si uno vive en Estados Unidos, a juzgar por la abundante información que se proporciona en libros o páginas web y que van desde establecimientos especializados hasta tiendas de jardinería

o bazares chinos donde, por lo que parece, es muy fácil encontrar una bola de este tipo. Doy fé de que por estas latitudes no es tan sencillo. Pero finalmente, después de muchos intentos fallidos, conseguí (en 2011) una buena bola especular, de unos 20 cm de diámetro, no muy cara (unos 40 €) de una tienda de decoración que me la envió por correo puntualmente. Su dirección es <http://decofactory.es/>. Espero que siga funcionando. Si no es así, la solución es armarse de paciencia y buscar por internet.

La bola puede situarse sobre una pequeña almohadilla, preferentemente de color gris neutro, rellena de arena para facilitar que se adapte a cualquier terreno y quede estable. Una alternativa más sofisticada y algo más complicada es montarla sobre un trípode sencillo. Esto tiene una ventaja adicional importante y es que nos permitirá nivelar la bola y conseguir que el horizonte quede perfectamente horizontal, algo que es importante para muchas aplicaciones. Pero puede ser complicado atornillar adecuadamente la bola al trípode. El libro de Christian Bloch sobre HDR (véase las referencias en la p 246 de la primera edición. Pueden encontrarse otras sugerencias similares por internet.

Una vez que se cuenta con todo esto, el procedimiento es el mismo que ya hemos visto en el apartado sobre grabación de series de fotografías con varias exposiciones. Es decir, habrá que encuadrar adecuadamente la bola, eliminar todos los automatismos (foco, balance de blancos, etc.), fijar la apertura a un valor tal como  $f/8$  o  $f/11$ , hacer una estimación del rango de velocidades necesario y disparar todas las tomas que haga falta para cubrir todo el rango dinámico.

Uno de los principales inconvenientes de este tipo de tomas es que aparecerá el reflejo del fotógrafo en el centro de la esfera. Una solución recomendable pero no siempre factible es tomar la fotografía lo más lejos que se pueda, forzando el zoom al máximo, para minimizar este defecto, lo que también es positivo pues se aumenta el campo reflejado. Y

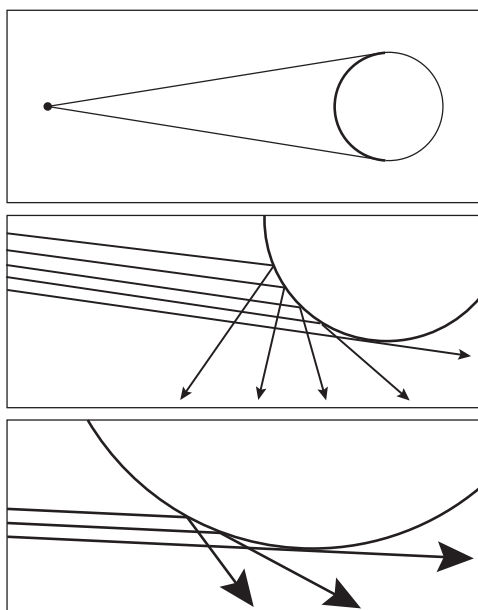


Figura 4.23 Campo visto (figura superior) y reflejado (figuras inferiores) que capta la cámara al utilizar una bola especular.





aceptar que el fotógrafo y su trípode son parte de la escena. Otra solución más complicada es tomar dos series, perpendiculares entre sí, colocando de nuevo la cámara y el trípode a unos 90° con respecto a la primera serie, procurando que la distancia y el zoom sean iguales, y utilizar la información de una de las dos series para clonar automáticamente la parte correspondiente en que no aparece el fotógrafo en un programa de edición, como HDR Shop, que cuente con esta opción. Otra solución menos complicada pero más inexacta es borrar el fotógrafo y su trípode en un programa de edición utilizando el pincel clónico y tonos cercanos. Veremos esta opción al ilustrar las posibilidades de edición en un programa que permita trabajar en 32 bits, como Photoshop, más adelante.

Hay que tener presente que partimos de requisitos contradictorios que no pueden solucionarse a la vez. Desde el punto de vista de la captación del mayor ángulo posible, lo que nos interesa es que la bola sea pequeña o que la cámara esté situada lejos. Desde el punto de vista de la resolución de la imagen que queremos obtener o desde el punto de vista práctico (puede ser que no tengamos espacio para alejarnos) lo que nos interesa es lo contrario. Así que habrá

que llegar a una solución intermedia entre estos extremos.

Por otro lado, también es importante tener en cuenta que, cuando se utiliza un programa de generación de panoramas para obtener un panorama equirectangular, a partir de la imagen captada por la bola especular, el programa asume que el campo captado por la bola es de 360°. Sin embargo, esto no es exactamente así. El ángulo es mayor de lo que parecería intuitivamente, como ya he mencionado y se muestra en la figura 4.24. Sin embargo, nunca llegará a abarcar 360° y en la práctica será de poco más que 320°. Esto quiere decir que la distorsión en los polos, que se traducirá en los bordes extremos del panorama, será considerable.

Por esta razón, sería deseable captar dos imágenes y componerlas pues, de este modo, reduciríamos el impacto de dos problemas importantes: eliminar el fotógrafo y su cámara y reducir (no es factible la eliminación total) la distorsión en los polos. Por desgracia, esto no es sencillo por lo que, en la mayoría de los casos, nos veremos obligados a introducir retoques manuales.

En cualquier caso, el resultado se tiene que procesar para obtener el HDR, con alguno de los métodos que se describen en el apartado



Figura 4.24 Reflectancia de una cartulina gris neutra y su reflejo en la bola especular. De los valores L correspondientes se deduce que la reflectancia de la bola especular es de un 81 % (0,47 / 0,58).





sobre Métodos de grabación y montaje de panoramas HDR a partir de escenarios reales.

### § § §

Las bolas especulares se utilizan también como medio de captar la distribución de luminancias de un entorno determinado. Si nos ceñimos al mundo real, hay varios modos de conseguir imágenes que cumplan estas condiciones. Todas tienen ventajas e inconvenientes que se compensan, por lo que no queda más remedio que describirlas todas.

Para utilizar una imagen HDR como un mapa de entorno en sistemas de simulación IBL (*image based lighting*), una de sus aplicaciones principales, que se desarrollará más adelante, necesitamos un mapa que cumpla dos propiedades importantes. La primera es que sea omnidireccional, es decir, que para cada punto de la escena haya un punto del mapa que se corresponda con esa dirección. La segunda es que el valor de ese punto se corresponda con un valor lumínico realista.

Para que se cumpla la primera condición, el uso de la bola especular es suficiente pues el entorno quedará representado con un ámbito aproximado de 180° en vertical y 360° en horizontal, como hemos visto.

Para que se cumpla la segunda condición necesitamos un dispositivo que almacene el rango de luminancias del entorno. A este tipo de capturas de la luz en un determinado escenario se les denomina genéricamente, *light probes*, que se puede traducir por “sondas de luz” o, más elaboradamente, como “recursos de indagación de la luz”. Estos métodos se utilizan para poder captar adecuadamente la luz que llega a un objeto determinado. Si, por ejemplo, se está filmando o fotografiando una escena desde diferentes puntos, la medición dada por el fotómetro de la cámara (o por un fotómetro manual utilizado desde la posición de la cámara), es equívoca pues mide la luz reflejada por el objeto, una medición que puede variar al cambiar el punto de vista. Por esta razón, los fotógrafos o sus ayudantes suelen desplazarse hasta la posición del objeto, tomar una medida de la luz incidente con un

fotómetro de luz incidente y anotar o vocear este dato al fotógrafo. Pero otro método que puede resultar más eficaz en muchos casos es utilizar una esfera que refleje la iluminación desde todas las direcciones y fotografiarla con dispositivos o métodos especiales que registren la iluminación real que recibe, con lo que se está registrando una determinada situación lumínica ligada a un momento y una zona determinadas. La ventaja de este método es que la captura queda grabada y puede utilizarse posteriormente.

De hecho, los profesionales utilizan no una sino dos bolas: una reflectante, que capte la distribución lumínica, y otra, pintada con gris mate de una reflectancia aproximada del 18 % (o una cartulina de gris medio estandarizado que se adquiere en cualquier tienda de fotografía), que proporcione una referencia adecuada para las superficies mates: un material virtual al que asignemos un color gris de las mismas características debería de representarse con el mismo color que esta bola de referencia.

En general, si se requiere precisión, es necesario utilizar este método para calibrar la reflectancia de la bola pues en la mayoría de los casos la esfera no refleja la luz perfectamente. Para saber con mayor exactitud su reflectancia se puede tomar una fotografía de una cartulina homogénea junto a la esfera y medir sus brillos respectivos. Dividiendo los dos valores se obtendrá aproximadamente la reflectancia de la esfera. La figura 4.25 ilustra este ejemplo. La medición, que se puede hacer con Photoshop o Gimp, de la zona central de la cartulina gris, nos da un valor RGB de 152, 135, 125 equivalente a Lab 58, 5, 8. Y la de su reflejo en la bola cromada, un valor RGB de 131, 106, 84 equivalente a Lab 47, 8, 16. Si nos basta con la medición de las reflectancias podemos quedarnos con los valores *L* que indican la luminosidad y deducir de estos valores la reflectancia de la bola cromada que será 0,81 (0,47/0,58). Si necesitáramos también conocer la desviación del color tendríamos que hacer la misma operación para cada uno de los tres canales.



### Grabación con cámaras especiales

El tercer método importante en la práctica es grabar imágenes HDR con un objetivo en ojo de pez (8 mm) y una cámara *full-frame* (con sensores de 24 x 36 mm). De este modo podemos captar directamente un campo de visión de 180° en horizontal y vertical apuntando al cenit. El principal inconveniente de este método es que un buen objetivo de ojo de pez es caro (más de 800 € en 2014) y una cámara *full-frame* todavía más (por encima de 1.500 € en esas mismas fechas).

Si se cuenta con un objetivo de este tipo, el procedimiento es el mismo que ya hemos visto: sacar varias imágenes con exposiciones variables y llevarlas a un programa de generación de archivos HDR. La cámara debe disponerse, prepararse y configurarse como también hemos visto (modo manual o con prioridad a la apertura, fijación del balance de blanco, uso de un trípode, etc.). Es importante asegurarse de que el trípode permite girar la cámara completamente de modo que apunte al cielo. Y que cuenta con un nivel para que quede horizontal.

Hay que tener en cuenta que si se va a apuntar al cielo, que es lo que nos interesa, el rango dinámico es muy alto por lo que habrá que hacer pruebas, observando cómo se desplaza el histograma, para averiguar los extremos a los que tenemos que llegar para cubrir todo el rango.

Una vez obtenidas las fotografías no estará de más editarlas en Raw para eliminar aberraciones cromáticas y comprobar que el balance de blancos es consistente.

### § § §

Una alternativa a todo lo anterior, que, por ahora, es prohibitiva pero con el tiempo es posible que adquiera mayor importancia, es obtener directamente imágenes HDR por medio de cámaras especiales. Dado que los avances en estos terrenos son continuos es previsible que en algunos años se pueda contar con cámaras digitales equipadas con sensores de mayor capacidad y que, por consi-

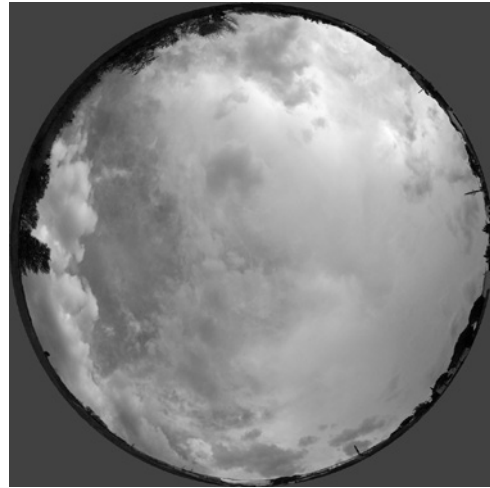


Figura 4.26 Escena fotografiada con una cámara *full-frame* y un objetivo de ojo de pez.

guiente, que los métodos que hemos descrito, de tomas de series con varias exposiciones, etc, sean innecesarios. De hecho ya existen cámaras de este tipo.

Una de las posibilidades más conocidas es la cámara SpheroCamHDR fabricada en Alemania, en Kaiserslautern, por la compañía SpheronVR ([www.spheron.com](http://www.spheron.com)). Esta cámara permite captar imágenes panorámicas con una resolución de hasta 13.000 x 5.300 píxeles y con un rango dinámico de 26 pasos EV que se procesan en un solo barrido. Permite obtener valores reales de luminosidad en un solo paso. El proceso puede durar de 1 minuto a 30 minutos, para una panorámica de 360° a plena resolución, según el rango dinámico que se necesite.

Sin embargo, el precio de estas cámaras es realmente alto (más de 50.000 € en 2010, aunque también se pueden alquilar por un coste, en las fechas citadas, de unos 800 € al día o unos 2.400 € a la semana) y está por ver su rendimiento real por lo que, como decía, estos párrafos no tienen otra intención que apuntar a un futuro más o menos cercano.

## Montaje y generación de HDR panorámicos

El primer método que he descrito se basa en la grabación de series de fotografías solapadas. Ya he mencionado que, para este caso, habría distintas alternativas para crear panoramas HDR:

1. Combinar primero cada grupo de imágenes homólogas (con la misma exposición) para crear series de HDR y luego combinar estos HDR en un panorama. La ventaja de este método es que la calidad final es mayor. La desventaja, que el proceso es algo más lento.
2. Montar panoramas a partir de las imágenes LDR y luego combinarlas en un HDR. La ventaja de esta vía es que la primera fase es más rápida y bien conocida. La desventaja es que el resultado puede ser impreciso pues las pérdidas de alineación de los LDR originales se transmiten a los HDR.
3. Hay una tercera alternativa que es sacar partido de los recursos incorporados a versiones más recientes de programas de *stitching* que permiten automatizar todo el proceso con bastantes garantías de que el resultado sea suficientemente exacto. Su principal desventaja es que, por ahora, no hay muchos programas corrientes que la admitan. Pero los que sí lo admiten son asequibles. Sin embargo, las pruebas que he realizado funcionan relativamente bien con PTGUI (la versión de pago) pero no tan bien con Hugin (la gratuita). El lector deberá decidir que es lo que le interesa y

adentrarse en los manuales de estos dos programas.

En la figura 4.27 (LDR), que es una variante de las incluidas en la ilustración del proceso previo, el proceso que he seguido fue el siguiente:

1. Llevar cada serie a Picturenaut (u otro programa similar). Como ya hemos visto, todo lo que hay que hacer es seleccionar cada secuencia y arrastrarla dentro de Picturenaut. Al cabo de unos instantes se genera el archivo. Guardarlo como EXR (o HDR si se va a llevar a un programa que solo admita este formato).
2. Llevar las fotos a Hugin (u otro programa similar) arrastrando las imágenes dentro del programa y comprobando que la distancia focal captada es de 8 mm y cambiando el objetivo a ojo de pez circular. No será necesario crear manualmente puntos de control si el solape es amplio, como lo era en este caso: bastará con pulsar el botón *Alinear*. Luego pulsar el botón *Crear la panorámica* (es lo mismo que ir al final, a la sección Empalmador y pulsar *Stitch* pero aceptando las opciones predeterminadas: revisarlas por si acaso). En principio esto debería ser suficiente.
3. Editar el resultado en Photoshop para corregir la proporción añadiendo una franja en la parte inferior para que la proporción sea equirectangular (2:1). Véase la última figura de las cuatro que se han incluido antes para ilustrar el proceso de captación y los comentarios que siguen.



Figura 4.25 Montaje de un panorama equirectangular a partir de series de fotografías solapadas y en formato HDR. En la imagen (LDR) se ha recortado la parte inferior para mostrar solo la parte principal que aparecería en la escena.



Como se puede comprobar en las figuras citadas, la proporción resultante es 2.5 aproximadamente. Estas imágenes finales tienen que retocarse por dos razones (véase más adelante los apartados sobre edición de formatos HDR en Photoshop).

En primer lugar hay, que comprobar que los 360° resultan continuos, desplazando la mitad de la imagen hacia un lado (utilizar el filtro de Photoshop que hay en el menú Filtros/ Otros/ Desplazamiento). Si hay pequeños defectos se pueden corregir manualmente (en la imagen final de las figuras citadas se han retocado algunos puntos del suelo en la parte inferior).

En segundo lugar hay que ajustar la proporción de unos 2,5, debida a que la imagen no abarca 180° en vertical. Para obtener un panorama de proporción adecuada hay que añadir una franja en la parte inferior, que no será visible cuando se utilice, como en la última imagen de la figura 4.21. Al añadir esta franja, el horizonte queda exactamente en el medio. De hecho, los panoramas HDR que se

comercializan por internet tienen a menudo todo el suelo borrado con lo que la imagen efectiva tiene una proporción de 4.0.

En la práctica solo se utiliza la mitad superior, que muestra el cielo. Según los casos puede ser necesario ajustar la posición o el tamaño del mapa para que se muestra algo de suelo. La figura 4.27 muestra el tipo de resultado con que se trabajaría en la práctica.

Hay una variante de este primer método que se basa en la obtención de 6 imágenes que corresponderían a las 6 caras de un cubo envolvente. Como este método se adapta mejor a la grabación de escenarios virtuales lo explicaré en la sección siguiente, Grabación y montaje a partir de escenarios vituales. Pero podría utilizarse también con escenarios reales, girando la cámara 90° en 6 direcciones perpendiculares entre sí. El proceso de montaje de estas 6 imágenes sería el mismo que se describe más adelante, en el apartado indicado.

§ § §

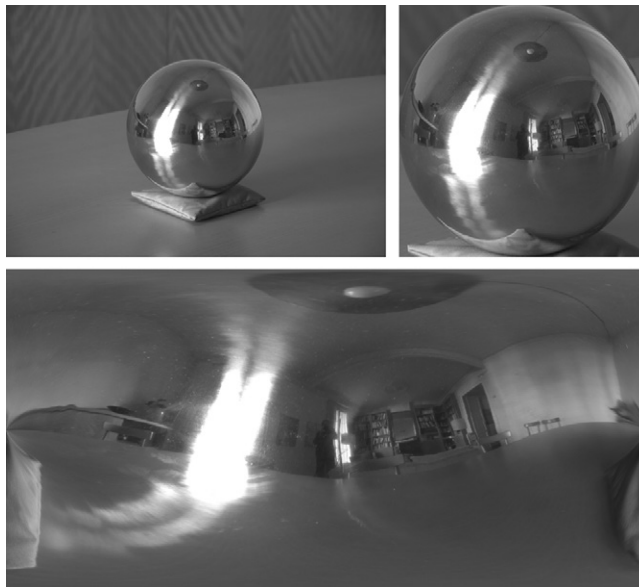


Figura 4.26 Panoramas HDR. Montaje. Bola especular (a) recortada (b) y desplegada (c) para obtener un panorama equirectangular.



El segundo método que he descrito permite obtener un archivo HDR a partir de series de fotografías de una bola especular y compuestas en un HDR único con un programa como *Picturenaut* u otro similar. En este caso, el montaje es muy simple pues todo lo que necesitamos es generar el HDR y luego recortar adecuadamente el resultado. Una de las opciones más sencillas es utilizar la versión 1.0 de HDR Shop que sigue siendo gratuita y se puede descargar de la página oficial (o buscar por Google con las palabras claves “hdr shop 1.0 download”). Suponiendo que se cuenta con este programa, que es muy sencillo de utilizar, bastará con hacer lo siguiente:

1. Desde HDR Shop 1.0 abrir el archivo HDR que hemos creado (menú *File/Open*).
2. Utilizar las herramientas *zoom* y *center* del menú *View* para visualizarla adecuadamente.
3. Ir a menú *Select/Draw Options/Circle*. Aparecerá un círculo sobre la imagen que se puede mover y cambiar de tamaño con facilidad. Ajustarlo cuidadosamente a la bola. Utilizar las herramientas de *zoom*, etc., para que el resultado sea lo más preciso posible pues en los bordes hay grandes distorsiones que se complicarán si recortamos demasiado o demasiado poco.
4. Ir a menú *Image/Crop*. La imagen se recortará limpiamente por los bordes de la esfera. El fondo permanece pero no se procesará al transformarse.
5. Guardar el resultado en formato HDR (la versión 1.0 de HDR Shop no incluye el formato EXR). Luego, si es necesario, convertirlo a EXR desde otro programa que lea ambos formatos.

De este modo, tenemos la imagen preparada para ser utilizada directamente o para ser transformada en otro tipo de mapa panorámico. La ventaja de utilizar HDR Shop es que podemos utilizar las herramientas de transformación, si nos interesan, para convertir directamente este resultado en un panorama o en un mapa cúbico. Véase la descripción de este

proceso que se da al final de esta sección. La figura 4.28 muestra la fotografía de una bola especular, la fotografía recortada en HDR Shop y transformada en este mismo programa para obtener un panorama equirectangular.

Si todo lo que queremos es recortar circularmente el mapa obtenido con la bola especular, podemos utilizar cualquier programa que permita trabajar con imágenes de 32 bits. Podemos utilizar *Photomatrix* aunque no lo hace mucho mejor. Y también *Photoshop* aunque la herramienta de recorte (*crop*) no muestra el círculo y para la transformación es necesario incluir un plug-in que no es gratuito (*Flexify 2*, de *Flaming Pear*). Otra alternativa es utilizar la herramienta de selección circular, ajustarla a la bola, invertir la selección y rellenar de negro.

§ § §

El tercer método descrito permite obtener un archivo HDR a partir de series de fotografías tomadas con un gran angular y compuestas en un HDR único por medio de un programa como *Picturenaut* o similar. Es decir, que el montaje es también muy simple y prácticamente idéntico al anterior. Si se utiliza HDR Shop 1.0 habrá que abrir el archivo, centrarlo y recortarlo y grabarlo siguiendo los mismos pasos que hemos descrito para el caso de la bola especular.

## Métodos de grabación y montaje a partir de escenarios virtuales

### Procedimiento general con Terragen

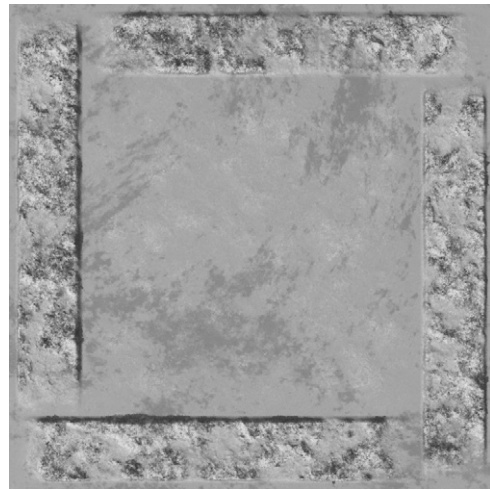
Se pueden generar panoramas en formato HDR desde programas de simulación que permitan guardar el resultado en este formato. Esto incluye a los principales programas de simulación, como *mental ray* (con *Maya*, *3ds Max* o *Soft Image*) o *V-Ray* o *Maxwell*. Pero si queremos que la simulación incluya paisajes y cielos con nubes las posibilidades son menores pues las capacidades de estos programas para este tipo de simulación son limitadas.





En 2014, una de las mejores opciones sigue siendo un programa como Terragen que ofrece una versión simplificada, gratuita y otra versión más completa (con mejores calidades de salida y mayor resolución) de pago. Es un programa relativamente sencillo de utilizar aunque está basado casi íntegramente en fractales, y aprender a controlar fractales para generar montañas, materiales o nubes no es trivial, pues hay que jugar con parámetros complejos, aleatorios, a los que se puede conducir en una determinada dirección pero sin que en ningún momento tengamos pleno control sobre los resultados. La notable calidad de los resultados finales se debe precisamente a esta autonomía “natural” que no podemos controlar plenamente. Pero en la mayoría de los casos, las opciones predeterminadas son suficientes para obtener el tipo de resultados que interesan para crear un paisaje virtual panorámico.

*Figura 4.1 No puedo incluir aquí una descripción completa del programa pues necesitaría muchas páginas. Lo que haré es describir de un modo muy sintético los métodos que he seguido para generar las imágenes que se incluirán más adelante. Pero es probable que estas explicaciones no se puedan seguir con facilidad si no se conoce mínimamente el programa. Recomendando, por tanto, al lector que quiera asimilar bien lo que sigue, que haga una pausa de un par de días para descargar una versión gratuita del programa, leer la ayuda y los tutoriales, que son más bien breves (una de las limitaciones del programa es que hay muchas funciones que prácticamente no están documentadas y solo pueden aprenderse buscando en los foros de internet, si hay suerte, o*



*Figura 4.29 Terreno generado con cinco Heightfield shaders: a) planta, b) vista general.*



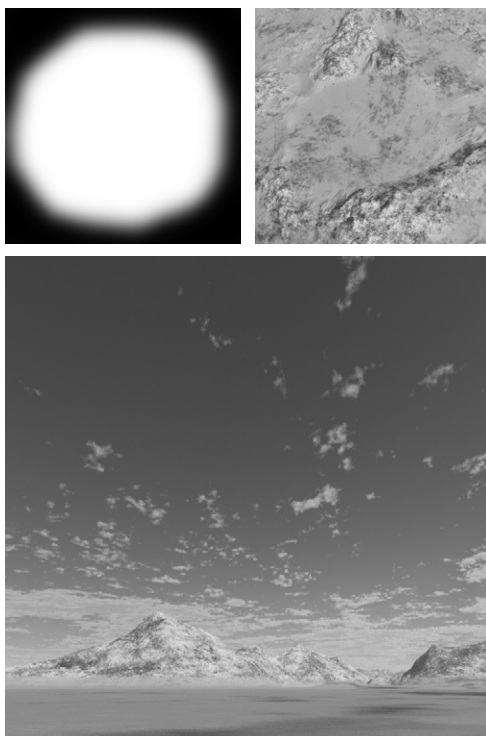


Figura 4.27 Terreno generado con un Power fractal y una imagen insertada: a) imagen, b) vista general, c) vista norte.

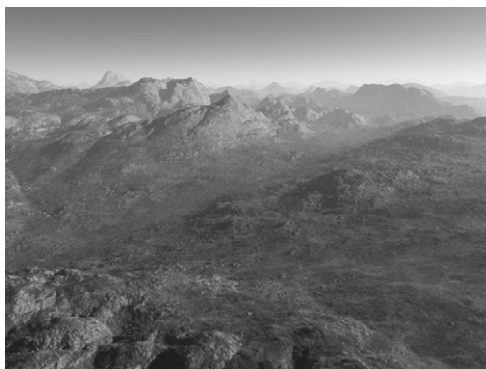


Figura 4.28 Ejemplo de la complejidad de resultados que se pueden conseguir combinando varios shaders de material.

haciendo pruebas) y crear algunos paisajes fractales.

Lo primero que se necesita es crear un terreno con un valle extenso, una zona central donde podamos colocar una cámara virtual con la que obtener imágenes en un ámbito de  $180^\circ$  en vertical y  $360^\circ$  en horizontal.

En Terragen podemos crear básicamente dos tipos de terrenos: un *Heightfield*, esto es, un terreno limitado a una zona rectangular especificada por el usuario, o un *Power fractal*, esto es, un terreno que cubre todo el planeta virtual en que se basa Terragen (y que tiene un radio igual al de la Tierra). En el primer caso, para crear un valle podemos crear cuatro *heightfields* dispuestos como en la figura 4.29 en torno a otro central. En el segundo caso, tenemos tres opciones: a) buscar una zona adecuada que haya surgido aleatoriamente para colocar nuestra cámara; b) crear un lago por medio de un recurso específico de Terragen que crea un plano a una determinada altura y luego cambiar sus características para que en lugar de simular agua, si esto no nos interesa, simule otro material; c) crear una zona plana por medio de una imagen añadida al fractal. En este caso, el procedimiento parte de crear una imagen en Photoshop o Gimp con la parte central blanca y los bordes negros y una transición suave entre estos dos tonos. Esta imagen se añade al *Power fractal* activando la opción *Blend by shader* y escogiendo como *shader* un *Image map shader* al que asignaremos esta imagen. Este último procedimiento es el que he seguido en las imágenes de la figura 4.30.

Lo siguiente que habría que hacer, aunque el orden es irrelevante (podemos ajustar la iluminación y las nubes y luego los materiales o al revés) es ajustar la iluminación. Esto es muy sencillo pues todo lo que hay que hacer, desde el panel *Lighting* de Terragen, es ajustar la dirección del Sol y comprobar que está activada la opción *Global illumination*, que lo está por defecto.

Para ajustar las propiedades del cielo y añadir nubes, ir al panel *Atmosphere*. Dejar los valores predeterminados, aunque puede



interesarse a revisar los parámetros correspondientes a cosas tales como la densidad de la niebla distante (*haze density*), su color en el horizonte (*haze horizon color*) o la intensidad y el color del cielo (*bluesky density*, *bluesky horizon colour*), etc. Si se quieren añadir nubes, presionar el botón *Add cloud layer* y escoger alguno de los tipos disponibles (*High level / Cirrus*, *Mid level / Altopcumulus*, *Low level / Cumulus*). Si se quiere controlar las características generales de las nubes, prepararse a dedicar algunas horas adicionales y comenzar por comprender bien el sentido de las variables principales: *altitude* (altura del centro de la nube sobre el nivel del suelo), *depth* (altura vertical de la nube), *coverage* (grado de cobertura de la nube) y, principalmente, los parámetros del fractal que genera la nube, el *Density shader* (al que se puede acceder directamente presionando el botón *Pattern*).

Lo siguiente que hay que hacer es incluir materiales para que el terreno resulte más interesante. Aunque es posible conseguir resultados notablemente realistas, como los que se muestran en la figura 4.31 (aunque está en blanco y negro por imperativos de edición: el original muestra una amplia variedad de tonos ocres y verdosos), en nuestro caso bastaría con soluciones relativamente simples. En la figura 4.32, que incluye las imágenes finales, se ha utilizado una combinación de cuatro *shaders*: “terreno” (el *shader* predeterminado *base colour*, con dos colores marrones, uno más oscuro y otro más claro), “hierba”, “montañas bajas” y “montañas altas”. Basta con asignar un color básico a estas capas y una función de color fractal que modifique este color básico aleatoriamente (presionar el botón “+” junto a *Colour function* y escoger *Create new shader! Colour shader! Power fractal shader*; luego acceder a este nuevo *shader* expandiendo el *shader* superior o volviendo a presionar el botón “+” y seleccionando “Go to Power fractal shader”). En la parte inferior, en *Coverage and breakup*, ajustar la cantidad de esta capa que se superpone a la anterior: moviendo el deslizador y comprobando cómo

varía este recubrimiento. Luego desplazarse a las secciones *Altitude constraints* y *Slope constraints*, al lado de la anterior, y comprobar cómo este recubrimiento varía al cambiar la altitud y la pendiente a que se aplica. Desplazar el cursor sobre la imagen que aparece en el visor y anotar los valores de la coordenada Y, que indica la altura y el valor de *slope*, para comprobar los valores adecuados. Quizás convenga utilizar el “test colour” para comprobar los resultados y luego desactivarlo. Si, por ejemplo, se especifica un valor de 25°, 85°, para la pendiente mínima y máxima, y un valor de 1, 10.000 para las altitudes mínimas y máximas, con zonas de transición en torno a unos 50 m, el resultado será que las partes más elevadas y más escarpadas se cubrirán del color especificado por este *shader*. Hay algunos ajustes adicionales pero espero que

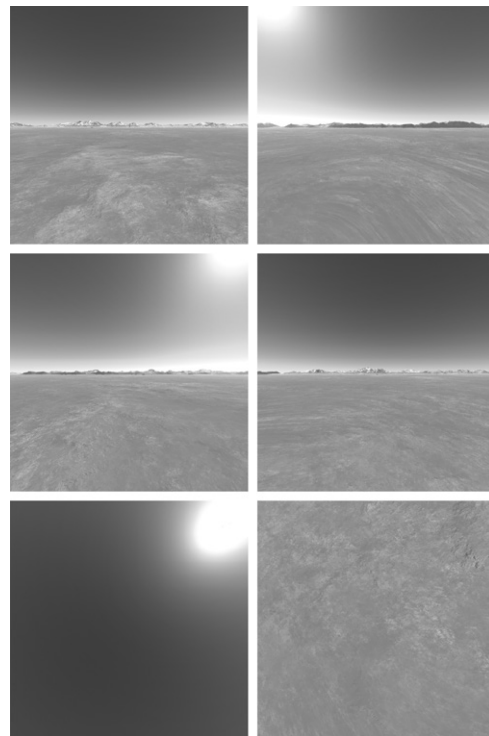


Figura 4.29 Las 6 imágenes tomadas desde las 6 cámaras. De izquierda a derecha y de arriba abajo: norte, sur, este, oeste, cénit y nadir.



esto será suficiente para que el lector se haga una idea del procedimiento.

El penúltimo paso es crear las cámaras que necesitamos para generar los panoramas, aunque a estas alturas habrá sido necesario crear al menos una cámara de trabajo para inspeccionar los resultados. Los mecanismos de control de las cámaras de Terragen son sencillos y algo menos sofisticados que los de programas corrientes, por lo que cuestan más de dominar y también es recomendable dedicar al menos media hora a familiarizarse con ellos, particularmente con los atajos de teclado y los valores numéricos.

Si se va a crear una *skybox*, se puede utilizar una configuración de cámaras como la siguiente:

1 *camNorte* (*Position*: 0, 200, 0, *Rotation*: 0, 0, 0, *Film aperture* / *FOV*: 36 x 36 / 90°, 90°.

El resto de las cámaras tendrán los mismos valores de *Position* y *Film aperture*. Todo lo que hay que hacer es copiar la anterior cinco veces (desde el *Node Network*, con Ctrl+C y Ctrl+V), y cambiar su nombre y sus coordenadas de rotación como sigue:

2 *camSur* (*Rotation*: 0, 180, 0), 3 *camEste* (*Rotation*: 0, 90, 0), 4 *camOeste* (*Rotation*: 0, -90, 0), 5 *camZenit* (*Rotation*: 90, 0, 0), 6 *cam-Nadir* (*Rotation*: -90, 0, 0).

Si los *shaders* de nubes, materiales y cámaras se van a utilizar en otros proyectos (otros terrenos), merecerá la pena exportarlos para no tener que repetir todos estos pasos. Para ello, seleccionarlos desde el *Node Network* y guardarlos como *clip* (menú *File/ Save Nodes As Clip Files*). Para insertarlos en otro archivo ir a menú *File/ Insert Clip File*. Estos archivos se guardan con la extensión *.tgc* a diferencia de los corrientes que se guardan con la extensión *.tgd*. Se debe tener en cuenta que en algunos casos, como los *shaders* de materiales, habrá que reconectarlos adecuadamente (para ello borrar el *Base shader* y conectar el primero a *Compute Terrain* y el último a *Planet*, reproduciendo la conexión del *Base shader*).

El último paso es configurar la salida de *Render*. Para ello, ir a la sección *Renderers*

donde nos encontraremos con dos salidas predeterminadas: *Quick render*, para pruebas y *Full render*, para la salida final. Especificar valores adecuados para ambas, que deberán ser cuadrados en nuestro caso (por ejemplo, 512 x 512 para pruebas y 1.200 x 1.200 para la salida final) y dejar los valores predeterminados de calidad, que son relativamente bajos en *Quick* y relativamente altos en *Full*.

Todo lo que queda por hacer es activar por turno cada cámara, asignarla en cada caso a la *Render camera*, activar el proceso y grabar el resultado en formato EXR.

Estas imágenes se pueden utilizar directamente para crear un *skybox* en 3ds Max o se pueden convertir en uno de los tres formatos mencionados al principio, tal como se explica en el apartado siguiente.

### Montaje externo

La salida de Terragen es en formato EXR. Puede ocurrir (ocurre con HDR Shop 1.0) que el programa no lea archivos EXR sino formatos HDR, como ocurre si se utiliza la versión gratuita de HDR Shop. Para convertir archivos EXR en HDR o en otros formatos similares (EXR half o TIF 32 bits), una opción sencilla y efectiva es utilizar el programa *Picturenaut*, ya mencionado, que también es gratuito y que permite grabar un archivo HDR como EXR. Y también permite guardarlo en formato EXR *half*, para ello escoger como formato de salida TIFF. Se abrirá un cuadro de diálogo, "Tiff save options", con tres opciones de profundidad de bits: 32 bits IEEE, 24 bits IEEE y *half* (OpenEXR).

En cualquier caso, partimos de la base de que contamos con 6 imágenes, que corresponderían a las 6 caras de un cubo sobre el que se proyecta el entorno.

Con estas 6 imágenes podemos hacer principalmente dos cosas: utilizarlas de modo independiente para crear una *Skybox* tal como se explica más adelante, en el apartado "Uso desde 3ds Max"; o bien utilizarlas para componer un mapa compuesto cúbico en cruz de alguno de los tipos que ya hemos visto.



Otra pregunta obvia es ¿para qué queremos un mapa de este tipo, cúbico, si en la mayoría de las aplicaciones que nos interesan vamos a utilizar las imágenes separadas o panoramas de otro tipo distinto. La respuesta es que, en ciertos casos, este tipo de mapas son más fáciles de generar y, a partir de ellos, se pueden generar los otros dos tipos de mapas, que son más difíciles de generar: los panoramas equirectangulares (*Latitude/Longitud*) y los angulares (*Light probe*). En el apartado correspondiente más abajo se explica como hacerlo.

Para montar las imágenes en cruz de modo manual habría que utilizar Photoshop o Gimp y colocar cada cuadro en su posición correspondiente. Como esto es un tanto pesado, puede merecer la pena hacerlo de modo automático. Para ello, se puede utilizar un programa sencillo que se puede obtener por internet, *cube2cross.exe*, que las une automáticamente. Este programa ha sido creado por Sean O'Malley (autor de varios plug-in para Terragen, agrupados bajo la denominación de *Sopack*) y se puede descargar gratuitamente (en 2012) de <http://www2.cs.uh.edu/~somalley/hdri.html>.

Una vez descargado este programa, todo lo que hay que hacer es asegurarse de que las imágenes están guardadas en una misma carpeta, y con nombres tales como *xxx\_ft.hdr*, *xxx\_bk.hdr*, *xxx\_dn.hdr*, etc., en donde "xxx" es el nombre que queramos pero lo que sigue es obligado, pues indica la orientación de cada una de las caras (*ft* es "front", *bk* es "back", *dn* es "down", etc). Luego hacer un doble clic sobre el archivo *cube2cross.exe*. Se abrirá una ventana en la que se pide seleccionar uno cualquiera de los 6 archivos y al cabo de unos instantes se generará un nuevo archivo con un nombre tal como *xxx\_cross.hdr* con los 6 archivos montados.

Por ejemplo, si en el ejemplo anterior, con Terragen, hemos obtenido una serie de archivos en formato EXR con nombres tales como *talycualNorte.exr*, *talycualSur.exr*, etc., hay que cambiar los nombres de modo que: a) el nombre base sea el mismo, es decir, en este caso, "talycual", b) el sufijo sea el indicado

más arriba. Es decir que habría que hacer los cambios siguientes:

*Norte*→*ft*, *Sur*→*bk*, *Este*→*rt*, *Oeste*→*lt*,  
*Cénit*→*up*, *Nadir*→*dn*

Y el resultado sería, por tanto:

*talycual\_ft.hdr*, *talycual\_bk.hdr*, etc.

Como habrá que abrir y volver a guardar los archivos, los cambios de nombre pueden hacerse cómodamente durante el proceso de conversión con *Picturenaut*.

### Procedimiento general con 3ds Max o programas similares

El procedimiento que sigue está basado en 3ds Max, pues los detalles quedarán más claros con un programa concreto. Pero puede

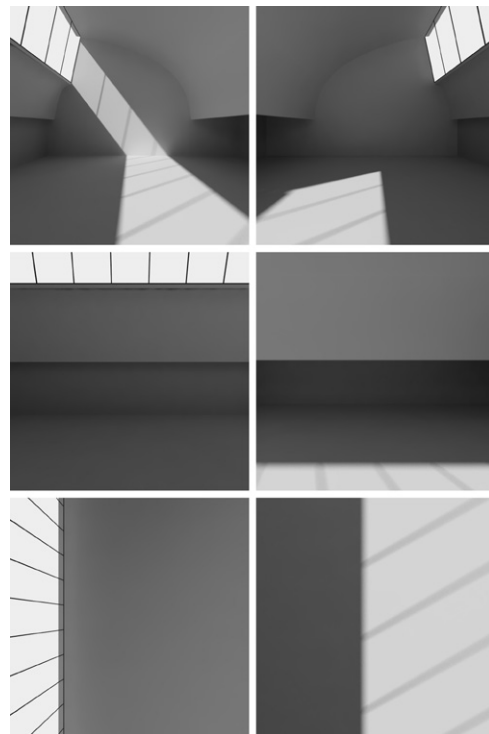


Figura 4.30 Las 6 figuras obtenidas por el procedimiento automático descrito en el texto. De izquierda a derecha y de arriba abajo: norte, sur, este, oeste, cénit y nadir.



utilizarse cualquier otro programa que cuente con recursos para generar mapas cúbicos y panoramas automáticos.

En 3ds Max podemos crear fácilmente un mapa cúbico mediante un material *reflect/refract* o crear directamente una proyección equirrectangular con la utilidad *Panorama exporter* o el *shader* de mental ray *WrapAround*. También puede crearse un mapa cúbico creando 6 cámaras de 90° de FOV, de modo similar a lo que hemos hecho en Terragen. Esto permitiría tener más control sobre los resultados pero no he encontrado ninguna ventaja significativa en comparación con los métodos automáticos.

En las figuras que siguen, 4.33 a 4.35, se han aplicado estos métodos a un interior con un techo curvo y un gran ventanal en uno de sus lados.

Para crear las imágenes de un mapa cúbico por medio de un material *reflect/refract*, el procedimiento sería el siguiente:

1. Colocar un objeto auxiliar (por ejemplo una pequeña esfera) en el punto desde el que se quiera generar los *renders* que, en principio, se supone que será en el centro de la escena.

2. Comprobar que las opciones de salida de *render* relativas a la calidad (*antialiasing*, etc.) son las adecuadas).
3. Crear un material *standard* y asignarlo al objeto. Asignar a *diffuse* un mapa de tipo *reflect/refract*.
4. Editar el mapa. Marcar la opción *Source: From File*. En *Size*, ajustar la resolución de salida al tamaño que interese. Escoger un valor que sea potencia de 2 (aunque esto no es necesario con las nuevas tarjetas gráficas, es probable que se procese mejor). En el ejemplo adjunto las imágenes tienen 1.024 x 1.024 píxeles.
5. En el grupo *Render cubic map files*, presionar el botón *To File*. En el panel que se abrirá dar un nombre y una ubicación para los archivos de salida. Especificar un formato EXR.
6. Presionar el botón *Pick object and render maps* y seleccionar el objeto auxiliar.

Tras unos instantes se generarán las 6 imágenes que habrán creado grabadas con el nombre y la ubicación que hemos dado. La figura 4.33 muestra estas imágenes para el ejemplo escogido.

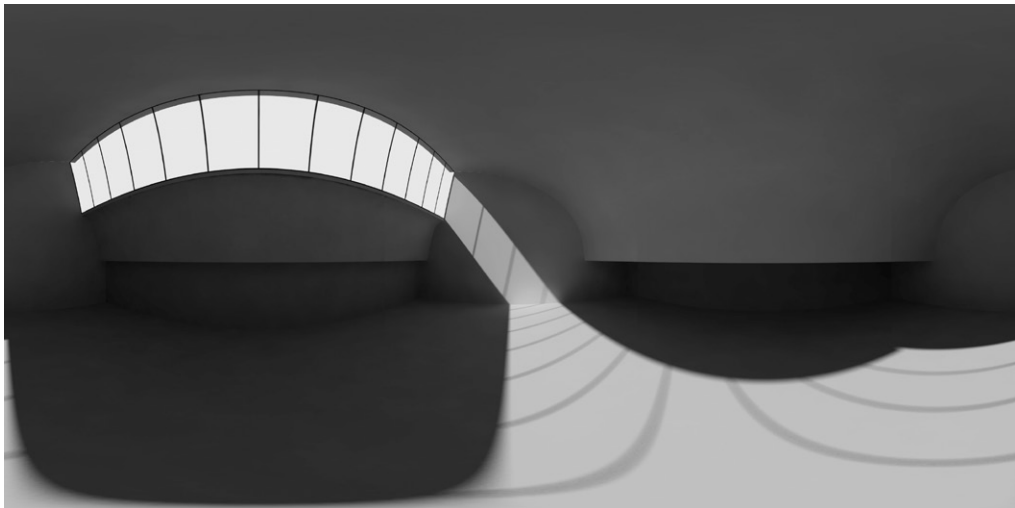


Figura 4.34 Panorama obtenido directamente desde 3ds Max.





Para crear directamente un mapa equirrectangular con el *Panorama exporter*, el procedimiento sería el siguiente:

1. Preparar un escenario adecuado. La figura 4.34 se basa en el mismo ejemplo anterior.
2. Situar una cámara en el lugar más adecuado para obtener un panorama del conjunto, que se supone que será, como antes, más o menos el centro de la escena.
3. Ir a *Utilities/ More/ Panorama exporter*. Presionar el botón *Render* con lo que se abrirá el *Render setup dialog*. Configurarlos así:  
En *Output size*, cambiar la resolución de salida a un valor alto, 2.048 x 1.024 por lo menos, mejor 4.096 x 2.048, aunque esto dependerá del caso y de la capacidad del ordenador.  
En *Render output*, hacer clic en el botón *File* y especificar una ruta, un nombre y un formato. Entre los formatos disponibles está EXR y HDR. Escoger uno de estos dos formatos. Véanse los comentarios sobre opciones que se dan posteriormente. Dejar el resto de valores tal como están.
4. Presionar el botón *Render* en la parte inferior de este panel. El programa hará 6 renders, uno para cada cara de un cubo virtual, y los combinará en un único archivo de proyección *Equirrectangular (Latitud/Longitud)* de proporción 2:1. Comprobar el resultado que deberá ser similar al de la figura 4.34.

También se puede visualizar el resultado volviendo a abrir el *Panorama exporter* y pinchando el botón *Viewer* con lo que se abre un visor especial. Localizar el archivo

que se ha creado y abrirlo. Con este visor se puede navegar por la escena. Y también se puede escoger guardarlo de nuevo y en otro formato. Los disponibles son *Cylinder* y *Sphere*. El primero da un resultado ligeramente más alargado en vertical y el segundo es el utilizado por defecto y da el mismo resultado que hemos obtenido.

5. El último paso es, como antes, guardar el resultado. Escoger igualmente EXR.

En principio, con 3ds Max tenemos más opciones, tanto con este procedimiento como con el anterior. Para ver la lista de opciones ir a la ayuda (*Image file formats/ OpenEXR*). Y otro tanto si se trabaja con otro programa.

Si se escoge HDR como formato de salida, se abrirá un diálogo que nos da dos alternativas: escoger la primera, "Use non clamped (real pixel) color channel", si no se va a utilizar mental ray o la segunda, "Use Standard RGB Channel", que guarda los datos en coma flotante, si se va a utilizar mental ray pues no todos los motores de *render* lo admiten (el estándar de 3ds Max, no). Esta segunda opción es preferible pues nos interesa, precisamente, trabajar con datos en coma flotante.

Si se escoge EXR como formato de salida se abrirá un cuadro con más opciones. Las que nos importan son las que aparecen junto a *Format: Half Float* (16 bits por canal) o *Full Float* (32 bits por canal). El resultado con *exr/16 bits*, que ya es suficiente, será igual al de HDR.

Si se prefiere utilizar el *shader* de mental ray *WrapAround*, el procedimiento tiene menos



Figura 4.35 Cuatro vistas de una escena vacía en la que se ha cargado el panorama anterior como fondo.





opciones que el anterior pero es más sencillo y cubre lo principal que nos interesa:

1. Situar la cámara en el centro de la escena o en una posición que se considere adecuada.
2. Ir a *Render setup/ Renderer/ Camera effects*. En el grupo de *Camera shaders*, activar *Lens* y presionar el botón adjunto. Seleccionar el *shader* de mental ray *WrapAround*. No hace falta editarlo porque no tiene parámetros: todo lo que hace es generar un *render* de 360° y lo adapta a las dimensiones de salida especificadas.
3. En *Render Setup/ Common*, especificar como tamaño de salida un formato de proporción 2:1 (hacer una prueba a tamaño pequeño, por ejemplo 800 px de ancho, bloquear la relación de aspecto y cambiar a un tamaño mayor, por ejemplo 2.048 o 4.096 de ancho.).
4. *Render*. El resultado será una imagen que abarcará 360° en horizontal y un ángulo en vertical que dependerá de la relación de aspecto.
5. Guardar el resultado y escoger como formato de salida HDR o EXR. Las opciones serán las mismas que hemos visto en el caso anterior.

### Montaje externo

Como ya he explicado en el apartado anterior, con los procedimientos descritos no hay que hacer ninguna operación adicional pues la imagen resultante ya es un panorama equirectangular listo para ser utilizado.

Si interesa, por alguna razón, convertirlo a alguno de los otros dos formatos, se puede utilizar el programa HDR Shop de modo similar a lo que ya hemos visto.

Si se necesita convertir este panorama equirectangular en 6 imágenes cuadradas para crear un *Skybox* también se puede utilizar otro programa gratuito, Cubing the Sphere que se podía descargar en 2012 de <http://darendash.home.comcast.net/~darendash/CubeTheSphere.html>.

El procedimiento de utilización de *skyboxes* y *skydomes* en 3ds Max se describe en el apartado correspondiente a IBL del capítulo 6 de este libro.

Para concretar y seguir con el ejemplo anterior, una vez que contamos con un panorama, obtenido por cualquiera de estos dos modos, para utilizarlo en una escena de 3ds Max como mapa hacer lo siguiente:

1. Abrir un nuevo archivo.
2. Cargar el panorama como mapa (tecla 8 o menú *Rendering/Environment*, para abrir el panel *Environment* y presionar el botón junto a "map").
3. Cambiar el tamaño de salida a un valor cuadrado (por ejemplo 512 x 512),
4. Hacer que el mapa se muestre en el visor (alt+b para abrir el panel correspondiente) y activar "mostrar marcos" (shift+F). El mapa aparecerá en el visor pero deformado para adaptarse al marco de salida.
5. Arrastrar el mapa al editor de materiales para ajustarlo. Cambiar la proyección a *Spherical*. Cambiar el *tiling* horizontal a -1.0 pues estos mapas quedan invertidos al utilizarse como mapas de entorno. Ahora el visor mostrará una porción sin deformar del panorama.
6. Si se modifican los valores de *U offset*, *V offset*, el mapa irá girando y podremos inspeccionar toda la escena. Y si creamos una cámara que apunte al centro de la escena y la vamos girando en torno a ese centro ocurrirá lo mismo.

Las imágenes de la figura 4.35 muestran el tipo de resultado que se obtendría en un archivo vacío, cargando el panorama como mapa, ajustándolo tal como se ha indicado más arriba, colocando una cámara en el centro de la escena y girándola para apuntar hacia las cuatro esquinas.



### 4.3 Edición y ajuste de imágenes HDR

#### Edición

##### Análisis previo

En primer lugar, después de haber generado un archivo HDR, sea con Picturenaut, Photomatrix, Photoshop o cualquier otro programa, conviene analizar el resultado para ver si es correcto. Para ello, necesitamos un programa que permita inspeccionar un HDR, pues lo que vemos en pantalla no es sino una de las posibles representaciones de una imagen que incluye más información de la que un monitor corriente puede mostrar, una versión LDR de esta imagen. Pero todos los programas citados incluyen un deslizador (*slider*) de ajuste de la exposición, que podemos mover hacia uno u otro lado para comprobar los valores extremos. O bien, como Photomatrix, incorporan un visor especial, una ventana que funciona como una lupa y nos permite inspeccionar el resultado. En Photoshop, al abrir una imagen de 32 bits, el menú inferior (abajo a la izquierda de la pantalla) mostrará automáticamente este *slider* (y si no fuera así, presionar la flecha que da acceso a diferentes opciones y escoger “exposición de 32 bits”). Tras desplazar el *slider* se puede hacer un doble clic sobre él para restaurar el valor original. Se debe tener en cuenta que estos cambios no afectan a la imagen sino solo a su visualización.

Sea como sea, conviene comprobar lo siguiente:

- El rango cubierto por la imagen, es decir, si en los extremos hay zonas que quedan recortadas antes de lo que interesa.
- El ruido y la calidad general de las zonas de sombra.
- El ruido y calidad general de las zonas de luces, en particular, si aparecen bordes excesivamente contrastados.
- La calidad de las zonas medias, principalmente en zonas de degradados en las que podrían aparecer saltos bruscos de gradientes.

- Defectos de alineación.
- Si hay figuras en movimiento que podrían haber pasado inadvertidas.

Si los resultados son insuficientes, tenemos dos opciones. Una es repetir todo el proceso. La otra es corregir manualmente los resultados si no son muy importantes. Algo que a menudo será prácticamente inevitable es corregir algún pequeño error, por ejemplo, borrar algún elemento indeseado en la imagen.

##### Herramientas y limitaciones de edición

Para editar un HDR la mejor opción, por ahora, es utilizar Photoshop pues es uno de los pocos programas que permite editar imágenes de 32 bits. Sin embargo, no todas las herramientas de edición corriente estarán disponibles. Por ejemplo, lo está el pincel clónico pero no el pincel corrector. Y lo está la varita de selección rápida pero no la varita de selección por valores. Y tampoco es un trabajo fácil pues habrá que ir comprobando el resultado, a medida que se trabaja, a diferentes niveles de exposición. Dicho de otra manera: tendremos que hacer dos tipos de *zoom*: uno, el corriente, para acercarse o alejarse a la imagen y otro, específico, para “acercarse” o “alejarse” hacia una u otra exposición, desplazando el *slider* a uno u otro lado. Así comprobaremos que saltos bruscos que son imperceptibles en un nivel son claramente perceptibles en otro.

También es importante entender como funciona el selector de colores, pues cambia de aspecto y de funcionalidad en 32 bits. Al abrirlo, su aspecto tiene poco que ver con el selector de color de Photoshop que se usa corrientemente. Si desplazamos el cursor fuera del panel y captamos un color de la escena, el color captado se mostrará en la parte superior como color nuevo (mientras que la parte inferior mantiene el color activo cuando se abrió el selector). Pero además, a cada lado de este color aparecen 3 colores más, etiquetados como -3, -2, -1 y +1, +2, +3. Estos colores muestran cómo aparecería este color con la exposición modificada según estos pasos.



Pasos que, por otra parte, se pueden cambiar pues hay otro parámetro “tamaño de paso...” que nos permite elegir entre 1 (la opción predeterminada), 2 (lo que daría -6, -4, -2 y +2, +4, +6) o 3 (lo que daría -9, -6, -3 y +3, +6, +9). Por añadidura, en la parte inferior aparece también un valor de intensidad expresado en “pasos”. Esto proporciona más opciones, pues podemos especificar un determinado color y luego aumentar o reducir considerablemente su intensidad, hasta extremos que solo se visualizarán si cambiamos la exposición global de la imagen.

Por otro lado, el formato HDR tiene una doble ventaja. En primer lugar, que el espacio de color (*gamut*) es mucho mayor, lo que permite representaciones más precisas; pero, en segundo lugar, a diferencia de lo que ocurre con LDR, las operaciones diversas a que puede ser sometido un formato (filtros, ajustes) no son destructivas. En un archivo con solo 255 niveles de luminosidad, los valores altos pueden ser recortados irreversiblemente por exigencias de algún tipo de ajuste que, por ejemplo, rebaje el color más blanco para igualarlo a otros blancos de la escena. En un archivo con varios miles de niveles, hay desplazamientos no destructivos que pueden revertirse.

## Ajustes

### Ajustes básicos

Hay ciertos ajustes que pueden ser necesarios en la mayoría de los casos. Lo que sigue es un resumen de los principales, referido a Photoshop (versión CS4, pero no muy distinto en las siguientes).

#### a) Recortar y enderezar la imagen.

Este proceso es similar al que se hace con cualquier imagen y también está disponible para HDR. Lo doy por conocido por lo que resumo muy brevemente los pasos característicos:

- i) Rotar la imagen para recuperar las verticales. Para ello, tomar una medi-

da con la herramienta regla de Photoshop y luego activar el comando Rotar (menú *Imagen/Rotación*) en modo “arbitrario” con lo que se abrirá un cuadro de diálogo con el ángulo correspondiente a la última medición. Confirmar. La imagen rotará este ángulo.

- ii) Enderezar las verticales. Activar la herramienta *Recortar* (*Crop*) en modo perspectiva o la herramienta *Recorte con perspectiva* en versiones posteriores a la CS4. Desplazar las asas del marco de recorte para que se adapten a líneas fugadas y luego confirmar. Las líneas fugadas se convertirán en verticales.
- iii) Recortar. Volver a ajustar el marco a la zona que se quiere conservar y hacer un doble clic dentro del marco.

#### b) Ajustar el equilibrio de colores.

Las herramientas de ajuste en este caso están más limitadas y se reducen prácticamente a las disponibles en el panel de ajuste de niveles.

Si la imagen está teñida de un tono cálido o frío y queremos recuperar su equilibrio, tenemos varias opciones que dependerán de los casos.

- i) Si hay alguna zona de la imagen que sabemos que tiene un color neutro podemos usarla como base para el ajuste. En este caso, abrir el ajuste de niveles (Ctrl+L o menú *Imagen/Ajustes/Niveles*). Luego escoger el *eyedropper* para grises y hacer pulsar el botón derecho del ratón para que se despliegue el menú que muestra el área muestreada. Según los casos, elegir la opción más adecuada, por ejemplo 3 x 3 o 5 x 5. Localizar en la imagen una pequeña zona que debería ser gris y hacer un clic. La imagen se modificará, pero el histograma quedará un tanto triturado. Sin embargo, esto que provocaría errores notorios en un archivo LDR puede aceptarse en un HDR.



- ii) Alternativamente, seleccionar el *eyedropper* para blancos y marcar una zona neutra. Esto hará que toda la imagen sea más brillante, pero puede reajustarse modificando la exposición. Para ello, tras este ajuste, abrir los ajustes de exposición (menú Imagen/Ajustes) y utilizar el *eyedropper* para blancos haciendo un clic en la zona más luminosa de la imagen, que volverá a reajustarse para adaptarse a este valor.
- iii) Una tercera alternativa es reducir directamente la influencia de un determinado color. Por ejemplo, supongamos que la imagen tiene una tonalidad claramente rojiza que queremos reducir. Abrir el panel de niveles (Ctrl+L). Seleccionar el canal correspondiente al rojo. Desplazar el cursor medio hacia la derecha con lo que su valor pasará de 1,0 a 0,9.... 0,8..., etc. Esto se traduce en una menor presencia de rojo, como puede comprobarse manteniendo abierto el panel de información que mostrará la composición RGB del punto sobre el que se sitúa el cursor.

Los ajustes de color deben llevarse a cabo con sumo cuidado pues si no se cuenta con suficiente experiencia y una buena vista hay más posibilidades de estropear la imagen que de arreglarla.

#### c) Ajustes de enfoque.

Photoshop cuenta con dos notables herramientas de enfoque que hacen milagros y funcionan también con imágenes de 32 bits. Probablemente se obtendrán mejores resultados con *Enfoque suavizado* aunque también puede probarse *Máscara de enfoque*. Ambas se pueden activar desde el menú *Filtros/Enfocar* y son las únicas que están disponibles con imágenes HDR. No abusar de la herramienta; es preferible quedarse un poco por debajo de los valores más altos de reenfoque, y comprobar el resultado variando la exposición como en los casos anteriores.

### Ajustes locales

Los ajustes locales se pueden llevar a cabo utilizando las herramientas corrientes de edición, con algunas limitaciones que se resumen brevemente a continuación y se ilustrarán con los ejemplos que siguen.

Se pueden modificar zonas con el pincel clónico. Como decía antes, el proceso es similar al que se sigue con imágenes LDR pero más complejo porque tenemos que hacer dos tipos de *zoom*: uno relativo al tamaño y otro a la exposición. La principal limitación es que no podemos utilizar el otro pincel clónico, el pincel corrector para suavizar el resultado final, por lo que habrá que ser más cuidadoso con las zonas de frontera que se crean al substituir píxeles.

Pueden utilizarse también las herramientas de suavizado local (gota, dedo) con las mismas precauciones de comprobar los resultados a diferentes niveles de exposición.

La herramienta de ajuste de color de Photoshop debe utilizarse con mucha prudencia pues la mayoría de los tonos de una escena son combinaciones de diversos primarios con lo que si, por ejemplo, corregimos un verde excesivamente brillante que nos molesta estaremos afectando a otros tonos que también incluyen un cierto porcentaje de verde. Con esta salvedad, puede merecer la pena, en algún caso, ajustar algún tono desde el panel Tono/Saturación (Ctrl+U o menú *Imagen/Ajustes*). En este panel escoger un color de la lista colgante que empieza por Todos. Al hacer esto, en la parte inferior se mostrará el rango correspondiente al tono escogido, un rango que se puede ampliar o reducir. Luego hacer alguna modificación de tonalidad (llevándolo hacia un tono cercano) o de saturación y comprobar el resultado manteniendo activada la opción de previsualización.

Para trabajar adecuadamente es prácticamente imprescindible utilizar máscaras de capa. Si el lector no está familiarizado con este recurso, la mejor recomendación es que haga una incursión en Google con las



Figura 4.36 Ejemplo primero: a) imagen inicial, b) máscara superpuesta con modificación de la exposición, c) retoques de la máscara, d) imagen final.

palabras clave “photoshop tutoriales máscaras de capa” y se encontrará con una buena colección. Resumo telegráficamente lo principal.

Para crear una máscara de capa, seleccionar la capa y presionar el icono correspondiente en el panel de capas (abajo, tercero por la izquierda). Aparecerá una segunda miniatura al lado de la de la capa, inicialmente cubierta de blanco. Las partes blancas muestran el contenido de la capa en que está la máscara. Las partes negras las ocultan y permiten ver el contenido de las capas inferiores. Por lo tanto, si activamos la máscara de capa y pintamos con negro hacemos “agujeros” en la capa y si pintamos con blanco los cubrimos. Y si pintamos con gris creamos veladuras, veremos la capa inferior con mayor o menor intensidad.

Si lo que está activada (marco resaltado) es la miniatura de la capa normal, pintar con estos colores tiene el mismo efecto que utilizar cualquier otro color, por lo que hay que tener cuidado y asegurarse de que está resaltada la máscara de capa antes de pintar con grises. Al resaltar la máscara de capa los colores de fondo pasan a ser grises, sean cuales sean.

Para conmutar con rapidez entre los colores de fondo presionar la tecla “x”. Y para ver el contenido de la máscara, hacer Alt+clic sobre su miniatura. Por último, si antes de crear una máscara de capa hay una zona seleccionada, el programa asume que la máscara de capa tendrá la forma de la selección. Y si lo que se crea es una capa de ajuste, la capa de ajuste incorporará esta selección como máscara de capa, lo que nos ahorra hacer lo mismo en varios pasos.

## Ejemplos

### Ejemplo 1

El ejemplo siguiente muestra cómo hacer ajustes locales de tono en una imagen HDR con Photoshop. Dando por supuesto que sabemos cómo crear máscaras de capa y otras



técnicas de Photoshop, el procedimiento básico sería el siguiente:

1. Seleccionar de modo aproximado la zona cuya exposición queremos ajustar.
2. Ir a menú *Capas/Nueva capa de ajuste/Exposición*. Dar por buenas las opciones del panel que se abrirá. Así aparecerá una nueva capa de ajuste sobre la que teníamos, con una pequeña máscara de capa al lado que coincidirá con la selección activa (que desaparecerá pues se substituye por la máscara).
3. Editar los parámetros de la capa de ajuste de exposición con *botón derecho ratón/Editar ajuste* sobre el nombre de la capa. Así se abrirá un panel, Ajustes (que también se puede abrir desde el menú *Ventana*). Este panel tiene tres controles: *Exposición* (modifica la exposición en pasos EV), *Desplazamiento* (oscurece o aclara las sombras y los medios tonos) y *Gamma* (modifica la distribución de los tonos aclarando u oscureciendo y aumentando o disminuyendo el contraste). Probar a modificar el primero para cambiar la exposición, desplazando los valores hasta recuperar los valores quemados o demasiado oscuros, según los casos.
4. Acercarse con *zoom* y retocar los bordes de la imagen pintando con blanco, negro o gris y utilizando las herramientas de suavizado para retocar las transiciones.
5. Comprobar los resultados en los extremos del rango modificando la exposición global de la imagen como hemos hecho en los casos anteriores.
6. Repetir el mismo procedimiento para otras zonas de la imagen. Cuando los resultados sean adecuados fusionar las capas. Completar el trabajo retocando las zonas de transición con las herramientas de suavizado local (gota de desenfoco, dedo).

La figura 4.36 muestra resultados parciales de este proceso cuya finalidad es recuperar el detalle de algunas zonas del patio que se



Figura 4.37 Ejemplo segundo: a, b) panoramas iniciales, c) proceso de superposición, alineación y desenmascarado, d) resultado final.

ve a través de la ventana y que no serían visibles por haber quedado sobrepuestas en una fotografía corriente.

## Ejemplo 2

En muchos casos nos encontraremos con que el resultado incorpora zonas muy desiguales, algunas de las cuales necesitan ser retocadas mientras que otras no. Esto es así





sobre todo en el caso de imágenes obtenidas a partir de bolas especulares y, en particular, a medida que nos acerquemos a los bordes en donde múltiples valores son comprimidos en cada vez menos píxeles, hasta llegar al propio borde, donde nos encontramos con puntos singulares de compresión infinita cuya información efectiva es nula.

No podemos hacer nada con estos puntos, que tampoco son tan importantes, pero sí con cosas como la imagen del fotógrafo y su cámara y su trípode.

Cada caso puede requerir diferentes tipos de intervenciones. Pero el ejemplo que sigue y que se ilustra en la figura 4.37, resume suficientemente el proceso básico.

Partimos de la base de que contamos con dos imágenes muy similares y que se prestan a ser alineadas por medio de lo que podemos denominar *manual stitching*.

1. En Photoshop, abrir las dos imágenes HDR y combinarlas en un único archivo (arrastrar la capa del segundo sobre el primero o seleccionar todo y copiar y pegar).
2. Ocultar y desocultar y cambiar la opacidad de la capa superior para ver la inferior. Utilizar guías que marquen la posición de elementos clave (esquinas de cuadros o ventanas, etc. Desplazar la capa inferior con la herramienta mover hasta encontrar una zona en la que se vea el fondo que habría detrás del fotógrafo y que queda oculta por este en la capa superior. Cuando se haya conseguido esta alineación volver a la opacidad original (100 %).
3. Crear una máscara de capa en la capa superior (icono correspondiente del panel de capas, tercero por la izquierda abajo). Activarla y pintar con un pincel suave y color negro sobre la figura del fotógrafo. Así hacemos “agujeros” en la capa superior y la figura del fotógrafo irá siendo substituida por el fondo correspondiente de la capa inferior. Si nos equivocamos, cambiar el color blanco por negro (un modo rápido de hacerlo

es presionando la tecla “x”) y pintar de nuevo para recuperar los píxeles de la capa superior. Basta con recordar que las partes negras de una máscara de capa son transparentes y las blancas opacas.

4. Acercarse y reducir el tamaño del pincel para retocar mejor los bordes. Utilizar la herramienta de suavizado para acabar de afinar las transiciones. Comprobar los resultados variando la exposición del HDR, como hemos visto más arriba.
5. Si la luminosidad de las capas fuera algo distinta, añadir una capa de ajuste de exposición a la superior (nueva capa de ajuste: cuarto icono por la izquierda, abajo, del panel de capas) y reducir o aumentar ligeramente la exposición.
6. Acabar de refinar el resultado, si fuera necesario, con el pincel clónico y refinando el resultado con la herramienta de suavizado.
7. Cuando el resultado sea satisfactorio acoplar la imagen y guardar el resultado.

### 4.4 Reproyección de valores (*tone mapping*). Ajustes de *gamma*. Control de exposición

#### Visualización LDR de archivos HDR

Un paso adicional requerido al crear archivos HDR es grabar una versión LDR del archivo para poder visualizarlo en un monitor corriente, pues el alto rango dinámico del archivo debe adaptarse al bajo rango dinámico del monitor, como ya he recordado varias veces.

Grabar un archivo LDR a partir de un HDR es un proceso de reducción. Debemos comprimir el rango dinámico de un modo adecuado para que se conserven los rasgos principales de la imagen. Es obvio que hay miles de posibilidades y que todas dependen de lo que consideremos como “adecuado”. Habrá que escoger algún tipo de algoritmo de los



muchos que han surgido en los últimos años a partir de los primeros que se comenzaron a desarrollar en la década de 1990 por Greg Ward, Paul Debevec y Erik Reinhard, entre otros. En el capítulo 2 ya he presentado un panorama general de los tipos de algoritmos y sus principios teóricos, así como la clasificación general que cabe hacer entre operadores TMOs (*tone mapping operators*) globales y locales. Aquí daré algunas explicaciones adicionales sobre su uso práctico y los pasos que hay que seguir para obtener una imagen por estos métodos.

Es un tema relativamente secundario para lo que nos importa, pero una breve revisión de los métodos de reproyección de la distribución de valores no estará de más. Entre otras cosas porque las ideas y técnicas son similares a las que se utilizan para otras cosas más importantes en la práctica, como el ajuste de *gamma* y el control de exposición.

### Reproyección de valores (*tone mapping*).

Los principios generales que se aplican en la distribución de tonos ya se han resumido

en el capítulo 2. Me remito, por tanto, a ese capítulo, recordando que no hay ni puede haber fundamentos objetivos para un problema similar al que abordan los pintores que tienen que representar la iluminación de una escena compleja con medios limitados.

Por un lado, deberán preservar la distribución general, pero en un rango de valores muy reducido. Por otro lado, deberán preservar el contraste de los detalles principales.

En última instancia, las decisiones se basarán en técnicas que combinarán recursos extrapolables a otros casos y recursos que dependen del caso singular de que se trate. Las técnicas digitales permiten automatizar estos procesos pero, en definitiva, tendrán que depender, en muchos casos, del juicio individual de quien las utilice.

### Métodos automáticos

#### a) Operadores globales.

Los operadores globales aplican una corrección de tonos a toda la imagen. Todos los programas de generación de archivos HDR cuentan con varios operadores que son ver-



Figura 4.31 Tone Mapping. Imágenes LDR obtenidas por la aplicación de diferentes algoritmos a una misma imagen HDR con Picturenaut (valores predeterminados):

a) Bilateral, b) Exposure, c) Adaptive Logarithmic, d) Photoreceptor.

Las diferencias pueden ser poco perceptibles en impresión sobre papel.



siones diversas de los principales que he resumido en el capítulo 2. Como en los casos anteriores daré tan solo un par de ejemplo, uno con Picturenaut y otro con Photoshop.

Desde Picturenaut, abrir una imagen HDR y hacer lo siguiente:

1. Activar *Tone Mapping* (menú *Image* o *Ctrl+T*).
2. Se abrirá un panel que presenta una pequeña imagen del archivo, un histograma en la parte inferior, una lista colgante de algoritmos junto a *Method* y los parámetros correspondientes al método seleccionado.
3. Escoger *Bilateral*. El método corresponde al presentado por Frédo Durand y Julie Dorsey (Durand y Dorsey, 2002). Incluye dos controles para ajustar el contraste y la saturación.
4. Escoger *Exposure*. Es un método básico que incluye dos controles para ajustar la exposición y el desplazamiento del histograma.
5. Escoger *Adaptive Logarithmic*. El método corresponde al método propuesto por Frederic Drago y que en otros programas se denomina por su apellido, Drago (Drago, 2003). Es similar al método de *Highlight Compression* de Photoshop. Pero a diferencia de este, que es automático, incluye cuatro controles para ajustar la exposición, el *bias*, la luminancia de las sombras y el contraste. Básicamente consiste en una compresión de los valores por medio de una distribución logarítmica pero preservando los valores altos.
6. Escoger *Photoreceptor*. El método corresponde al método propuesto por Eric Reinhard (Reinhard, 2002). Incluye cuatro controles para ajustar la exposición, la compresión dinámica, la saturación y el contraste. La idea principal es emular la respuesta del ojo humano por medio de una curva en forma de S que proporciona una distribución logarítmica para los valores medios altos y medios bajos, que preserva hasta cierto punto los valores extremos.

7. Guardar el resultado en alguno de los formatos LDR disponibles (tga, jpg, png, ppm y tif).

La figura 4.38 muestra diferentes resultados de aplicar estos cuatro métodos a una misma imagen. Las diferencias son sutiles y puede que no resulten muy notorias en una reproducción impresa en blanco y negro por lo que recomiendo al lector que reproduzca el proceso por su cuenta y analice los resultados en un monitor bien calibrado.

Desde Photoshop, abrir una imagen HDR y hacer lo siguiente:

1. Ir a menú *Imagen/Modo* y escoger 8 bits/canal. Al escoger este modo se abrirá un cuadro especial, Conversión de HDR. Este cuadro ofrece cuatro métodos posibles de conversión (Exposición y *gamma*, Compresión de iluminación, Ecuilizar histograma y Adaptación local). Los tres primeros son métodos de ajuste global de tonos; el cuarto se verá más adelante.
2. Escoger Exposición y *gamma*. Hay dos parámetros disponibles, los dos que dan nombre al método. Su sentido es bastante obvio. Si se aumenta o disminuye el primero, exposición, la imagen se aclarará o se oscurecerá como si modificáramos la exposición en la propia cámara. Si se aumenta o disminuye el segundo, *gamma*, el efecto será disminuir o aumentar el contraste.
3. Escoger el segundo, Compresión de iluminación (*Highlight Compression*). Es similar al método *Adaptive Logarithmic* de Picturenaut y como este, parece estar basado en el método propuesto por Frederic Drago. En este caso no hay opciones. Con este método se preservan los valores altos y se comprimen los restantes por medio de una distribución logarítmica. Aunque no haya opciones, queda la posibilidad de introducir correcciones globales por medio de los controles característicos de Photoshop, manipulando el mapa de curvas y volviendo a grabar el resultado.



4. Escoger el tercero, Ecuilizar histograma. En este caso tampoco hay opciones. El algoritmo es puramente automático y consiste, básicamente, en una reducción de los valores del histograma manteniendo las relaciones globales, lo que tiene el efecto de aumentar, en general, el contraste. Tiende a juntar excesivamente los valores en los extremos con lo que se pueden perder matices en las zonas iluminadas y en las sombras.
5. Tras escoger el ajuste más adecuado guardar el archivo en el formato LDR que interese.

Los resultados serán similares a los obtenidos con Picturenaut y que se han ilustrado en la figura 4.38: ligeras diferencias que se apreciarán mejor en un monitor que en una impresión sobre papel.

#### b) Operadores locales.

Los operadores locales permiten actuar sobre zonas específicas de la imagen. En Photoshop podemos escoger esta opción como una alternativa específica y ajustarla desde el mismo cuadro de diálogo que los globales. Sin embargo, como veremos, es preferible utilizar otros métodos. Para probar esta alternativa hacer lo siguiente:

1. Ir a menú *Imagen/Modo* y escoger 8 bits/canal. Al escoger este modo se abrirá el cuadro Conversión de HDR. Como ya hemos visto, este cuadro ofrece cuatro métodos posibles de conversión (Exposición y *gamma*, Compresión de iluminación, Ecuilizar histograma y Adaptación local). En los párrafos anteriores hemos visto los tres primeros.
2. Escoger el cuarto método, Adaptación local. En este caso, las curvas quedan disponibles y podemos modificarlas para afectar directamente al contraste y la luminosidad del modo habitual. Desplegar la parte inferior del cuadro de diálogo para que se muestren el cuadro de curvas e histogramas. Analizar la escena pasean-

do el cuentagotas (*eyedropper*) sobre ella y haciendo un clic para que se muestre el valor del punto seleccionado sobre la curva. Después de analizada, marcar algunos puntos, marcar la opción “vértice” para que el desplazamiento del punto no afecte a toda la curva y desplazar puntos para variar el contraste local.

3. Después de haber hecho unos cuantos ajustes, modificar los parámetros Radio y Umbral. El primero, Radio, afecta a la extensión del área sobre la que se aplicarán los cambios. Un radio de 1 px significa que solo se afectará a 1 píxel y a sus vecinos inmediatos, es decir, un total de 5 píxeles. El radio adecuado dependerá de la resolución de la escena y de los detalles que nos interese realzar. El segundo, Umbral (*Threshold*) especifica la distancia máxima de luminancia dentro de una determinada zona. Si el valor es muy bajo, el área sobre la que se aplicará el ajuste es muy pequeña, puede llegar a ser de un solo píxel, con lo que la corrección se aplicará indiscriminadamente a todos los píxeles y el resultado será similar a aplicar un operador global. Si es muy alto, aunque las diferencias de luminosidad sean muy altas se aplica el mismo tratamiento: el resultado puede ser que aparezcan halos en algunas zonas. Un criterio que puede servir es forzar el umbral hasta que aparezcan halos y luego bajar el valor.

Sin embargo, como decía anteriormente, el mayor control local se consigue mediante ajustes manuales, es decir, mediante un operador local humano. Esto da más trabajo pero también permite obtener resultados que serían imposibles con métodos automáticos. Me remito, por tanto, al ejemplo dado más arriba, el ejemplo 2, donde parte de las operaciones que se resumían pueden ser consideradas como métodos de ajuste de tonos local.



## Ajustes de gamma

Lo que sigue afecta solo a algunos casos especiales, poco frecuentes. Pero, si se diera la circunstancia, esta sección puede ayudar a entender mejor los problemas involucrados.

### Tipos generales de ajustes

La corrección de gama se aplica a todas las imágenes digitales (a los monitores y dispositivos de salida en general) para compensar por la no linealidad de la distribución de intensidades. Esto ya se ha visto en el capítulo 2 y también hemos visto que plantea un problema de difícil solución pues las redistribuciones en unos casos pueden ser inadecuadas en otros. Los monitores actuales (2010) están ajustados a una corrección de gama de 2,2 que es adecuada para las imágenes corrientes, es decir, para las imágenes en color real que se basan en el modo RGB con una profundidad de color de 8 bpp o 256 colores.

Sin embargo, este ajuste no sirve para imágenes de alto rango dinámico. En principio esto no supone un problema pues las imágenes HDR se utilizan internamente, con una gama lineal de 1.0. Y si se convierten a imágenes LDR, para ser usadas externamente, el programa de conversión ya se encarga de la corrección de gama.

Pero tampoco sirve para cálculos de iluminación que utilicen intensidades reales muy altas, como ocurre en una escena diurna.

Así que, de entrada, nos encontramos con dos tipos generales de ajustes que, en algún caso, por suerte no demasiado frecuente, como decía al comienzo, habrá que utilizar:

- a) Ajustes generales que dependen de Windows y que afectan a todos los programas utilizados. Partimos de la base de que esto ya se ha hecho, es decir, que contamos con un monitor adecuadamente calibrado. Pero no debe perderse de vista que en muchos casos esto puede no ser así, lo que afectará a todos los ajustes que vienen a continuación.

- b) Ajustes que dependen de un programa determinado.

En este último caso nos encontramos con diversas posibilidades que varían con el programa utilizado. En un programa de simulación visual, como, por ejemplo, 3ds Max, nos encontraremos con las siguientes posibilidades características:

1. Ajustes que afectan a toda la escena. En el caso de 3ds Max estos ajustes se controlan desde el cuadro de diálogo *Gamma and LUT* (menú *Customize/Preferences*). Para comprobar su funcionamiento, activar la casilla *Enable gamma* y modificar el valor de *gamma*. Probar a aumentarlo (por ejemplo a 4,0 o más) y comprobar que tanto los colores de la escena como los de un *render* (mantenerlo abierto) se aclaran. Probar a disminuirlo (por ejemplo a 1,0 o menos) y comprobar que los colores se oscurecen. Si se marca la casilla *Affect color selectors*, los selectores de color también se modifican para reflejar estos cambios. Dejarlo en su valor original (2,2) a no ser que haya buenas razones para cambiarlo (el cuadrado gris de muestra debe tener una intensidad similar a la de su fondo y similar a la intensidad global del monitor).
2. Ajustes que dependen del cálculo de iluminación. Estos ajustes se gestionan con el control de exposición. Ver el apartado correspondiente a estos ajustes más adelante.
3. Ajustes que afectan a los materiales, principalmente a los mapas. Este último caso es algo más complejo y lo discuto con algo de detalle a continuación aunque,afortunadamente, se presenta pocas veces en la práctica.

### Ajustes locales de gamma

Para empezar, el mapa de bits puede tener su propio ajuste de gama, aunque esto no sea muy corriente. Al asignar un mapa de bits a



un material, cuando se abre desde el explorador, en la parte inferior del cuadro de diálogo hay tres opciones: a) utilizar la gama propia del mapa, b) utilizar la gama predeterminada por el sistema (esta es la opción por defecto y la preferible), c) substituir (*override*) la gama del mapa. Si se quiere saber si el mapa tiene una determinada gama insertada, presionar el botón *info*. En el caso de un mapa de bits TGA se encontrará que sí la tiene y es de 1,0. Pero si se utiliza la segunda opción (*gamma* del sistema) el resultado será idéntico que si se utiliza la primera pues el sistema ya corrige las particularidades de cada formato por lo que tampoco tenemos que preocuparnos de este caso.

En el panel *Gamma and LUT* (menú *Customize/Preferences*), activar la casilla *Enable gamma* y, en el grupo *Bitmap files*, modificar los valores de *Input gamma* (2,2) y *Output gamma* (2,2). Si, por ejemplo, se baja el valor de input a 1,0 el resultado será más claro y si se aumenta a 4,0, más oscuro. Pero esto solo afectará a los materiales con mapas de bits y no a los materiales que solo tienen color asignado por parámetros básicos.

Para que estos ajustes no afecten a todos los mapas de la escena, lo más recomendable será volver a abrir el archivo y especificar un valor de *gamma* de entrada (*input gamma*) adecuado solo para ese mapa. Pero es mejor utilizar las alternativas que se dan más adelante.

Estas configuraciones se conservan con la escena. Puede ocurrir que se abra una escena con configuraciones de gama interna que no coincidan con las establecidas en la sesión activa. En este caso aparecerá el mensaje "The Gamma & LUT settings of the file do not match the System Gamma & LUT settings".

En general, es preferible no tocar la configuración general y, si es necesario, ajustar el color de un material particular, sea directamente, sea mediante el mapa especial de *mr Gamma & Gain* que se explica a continuación. Pero conviene tener en cuenta todo lo anterior pues puede ser que en algún caso

lleguen a la escena mapas de bits con gamas propios y nos encontremos con estos avisos.

Los mapas HDR de 32 bits no funcionan bien con los ajustes preestablecidos. En estos casos suele ser preferible establecer su *gamma* en 1,0 (con la opción *override*, al abrirllos). Los mapas HDR de 16 bits pueden requerir este ajuste o no, según los casos.

Los mapas que vayan a utilizarse para efectos de relieve deberían también reajustarse a 1,0 para que los valores de desplazamiento no se aumenten en los tonos medios lo que alteraría artificialmente el resultado en algunos casos (no es necesario hacerlo si solo van a contar los extremos, blanco y negro).

En la práctica no será necesario hacer cambios en la gran mayoría de los casos, pero he incluido este apartado para que se entienda el sentido general de avisos que aparezcan con cierta frecuencia y porque, en algún caso especial, podría ser necesario modificar valores de archivos de entrada en un programa de simulación.

## § § §

Tampoco está de más mencionar que, cuando se utiliza mental ray, el mapa (*shader*) *mr (mi) Gamma & Gain* permite controlar el ajuste de gama por material lo que es claramente preferible, pues los cambios solo afectarán a un determinado material, no al conjunto de la escena. En otros programas se pueden encontrar recursos similares. Para utilizarlo hacer lo siguiente:

1. En el editor de materiales, seleccionar el material con mapa que se quiere ajustar. Bajar al nivel de mapa y pulsar el botón *Bitmap*. En el navegador que se abrirá, escoger, en mapas de mental ray, el mapa *Gamma & Gain*.
2. Responder sí, a la pregunta de si se quiere conservar el mapa previo como submap ("Keep old map as submap?"). El nuevo mapa "envolverá" al previo.





3. Este mapa solo tiene 4 parámetros y solo hay dos que importen realmente. El primero, *Input*, muestra un color, predeterminado a negro, que puede cambiarse y se superpondría al color del material si no utilizáramos un mapa, lo que, en general, no interesa pues este *shader* se utiliza para corregir el *gamma* de mapas. El cuarto, *Reverse gamma correction* está activado y es mejor dejarlo así para que aplique una corrección inversa a la entrada y luego la divida por el valor de *Gain*. Si se desactiva lo hace al revés (primero lo multiplica por *gain* y luego le aplica una corrección *gamma*).
4. Modificar el segundo parámetro, *Gamma*. Al reducirlo, el mapa se aclara en las zonas medias. Y a la inversa.
5. Modificar el tercer parámetro, *Gain*. Al aumentarlo, el mapa se oscurece uniformemente. Y al disminuirlo, el mapa se hace más luminoso.

Como en los casos anteriores, estos ajustes se utilizan raramente en la práctica pero los menciono por dos razones. La primera, porque en algún caso especial puede ser interesante saber que existen. Por ejemplo, como veremos más adelante, es una de las posibles opciones (aunque no la más recomendable) para solucionar problemas con fondos demasiado oscuros o demasiado claros. La segunda, porque se relacionan conceptualmente con otros temas de mayor importancia en la práctica, como el que se trata en el apartado siguiente.

### Control de exposición

Cuando se utilizan sistemas de simulación de iluminación avanzada con luces que simulan intensidades reales como ocurre, por ejemplo, en los sistemas de iluminación de luz diurna, los valores de intensidad son muy altos, del orden de 80.000 a 100.000 luxes. Pero los monitores están calibrados para intensidades de bajo rango dinámico, con valores convencionales comprendidos entre 0 (negro) y 1

(blanco), y con una curva de distribución que hace que las intensidades intermedias se escalonen de un modo perceptivamente regular. Al introducir valores muy altos, esta distribución se altera y, si no se introduce algún tipo de corrección, el resultado de activar el comando *render* será que la escena aparecerá quemada, con todos los valores distorsionados, excesivamente luminosos.

Para solucionar este problema la mayoría de los programas de simulación incorporan un recurso denominado control de exposición que redistribuye los valores. El recurso puede estar incorporado a una cámara física (*physical camera*), como ocurre en programas como V-Ray o Maxwell, o ser independiente de la cámara, como ocurre con 3ds Max y mental ray.

Esta redistribución es similar a la que se efectúa con *tone mapping*, si bien en este caso no se trata de adaptar un archivo HDR a un sistema de visualización LDR sino de adaptar una distribución de valores derivada de un cálculo a un sistema de visualización LDR.

Los programas más simples incluyen métodos de corrección logarítmicos que consisten en redistribuir los valores obtenidos durante el cálculo por medio de una curva similar a la que usa el propio monitor para adaptar los valores corrientes. Pero los métodos de este tipo son muy rígidos y no se adaptan bien a los diferentes casos. Los programas más avanzados incluyen métodos de corrección más sofisticados que se basan en simular el comportamiento de una cámara fotográfica.

Así, del mismo modo que con una cámara fotográfica adaptamos la exposición a las características de la escena, dejando que entre más luz si la escena es más bien oscura y, a la inversa, reduciendo la entrada de luz si la escena es muy luminosa, estos controles permiten adaptar la salida de *render* a las características de la escena con parámetros que simulan las características de una cámara.

Con mental ray, virtual ray o maxwell, por no mencionar sino alguno de los programas más utilizados, se puede utilizar un sistema de



este tipo. Con mental ray, por ejemplo, queda disponible un recurso denominado *mental ray photographic exposure control*. Para utilizarlo, desde 3ds Max ir a menú *Rendering/Exposure control*, con lo que se entra en un panel que incluye una lista colgante con cinco alternativas: *Automatic*, *Linear*, *Logarithmic*, *Pseudo color* y *mr Photographic exposure control*. Las dos primeras se mantienen por compatibilidad o para casos especiales que no nos interesan. La tercera, logarítmico, se utiliza en sistemas de iluminación avanzada distintos de mental ray (por ejemplo, *radiosity* o *light tracing*) y, como he dicho, es bastante limitada. La cuarta, *Pseudo color*, se utiliza para visualizar las intensidades por medio de colores puros: azul, cian, verde, amarillo, naranja y rojo, de modo que el azul corresponde a las zonas más oscuras y el rojo, a las claras. Esto se utiliza para facilitar el análisis de los resultados pero no afecta a la iluminación de la escena. La quinta es la que proporciona mayor control y de la que resumo lo principal a continuación.

Con *mr Photographic exposure control* tendremos controles que simulan los de una cámara fotográfica para ajustar la exposición. Puede utilizarse modificando el *Exposure value* (EV, Valor de Exposición) o cambiando la velocidad de obturación, la apertura o la velocidad de la película. El resultado es el mismo y, si la escena es fija, no tendrá mucho sentido utilizar diferentes combinaciones de apertura y velocidad de obturación por lo que resultará más sencillo introducir un valor directamente para el parámetro EV.

Se pueden utilizar configuraciones predefinidas descolgando la lista junto a *Preset*. Las disponibles son las siguientes: a la derecha indico los valores de EV:

Exterior, luz diurna, cielo claro:	15
Exterior, luz de noche:	3
Interior, luz diurna:	10
Interior, luz nocturna	2

En la práctica, lo más recomendable es partir de alguno de estos valores predeterminado y ajustarlo en función del tipo de escena. Así,

por ejemplo, podemos partir de un valor de EV 10 para una escena interior con luz diurna pero lo más probable es que tengamos que movernos en un rango de 8,0 a 12,0, según los casos, o acabar adaptando un valor intermedio, por ejemplo, 11,5. Tan solo las características de la escena y las pruebas que hagamos nos dirán cuál es el valor adecuado.

El resto de los controles son similares a los del control logarítmico aunque las unidades

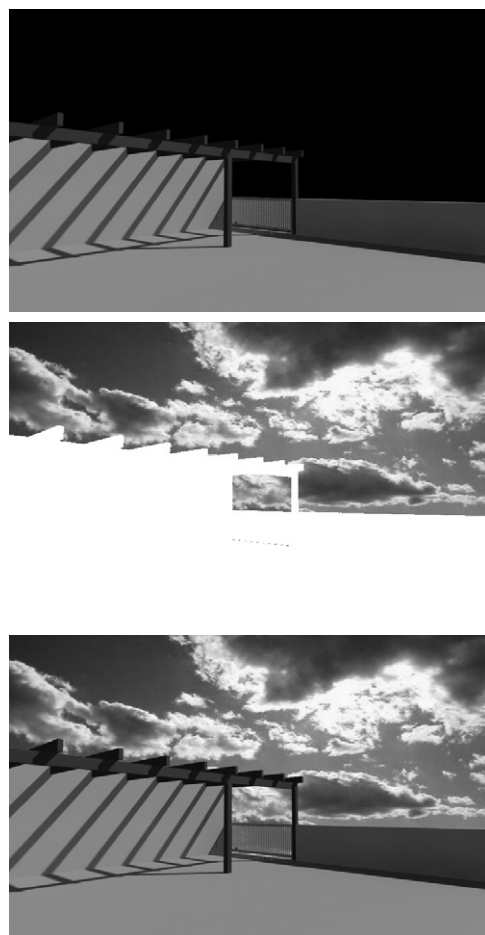


Figura 4.32 Fondos demasiado oscuros o demasiado claros en una escena con iluminación avanzada, rangos dinámicos reales y un fondo LDR: a) Process backgrounds activado y EV 15, b) Process backgrounds activado y EV 0, c) Process background desactivado y EV 15.



son distintas y se pueden utilizar para modificar a posteriori la imagen de salida. *Highlights* (0,25), *Mid Tones* (1,0) y *Shadows* (0,2). También hay ajustes para modificar la Saturación de color (1,0), la temperatura de color del punto blanco (6.500°K) y el viñeteado (0,0). La escala física se mide en  $\text{cd/m}^2$ . El control de exposición utiliza un *gamma* propio de 2.2 para prevenir desajustes con el del monitor, el mismo que hay por defecto en *Preferences/Gamma and LUT*. Conviene activar la corrección de *gamma* con el control de mr y desactivarlo (pues es básicamente el mismo) con Logarítmico. De todas formas, estos ajustes suelen venir dictados por los resultados que estamos obteniendo.

### Problemas con fondos y control de exposición

Hay algunos problemas característicos que pueden aparecer cuando se utilizan mapas para el fondo:

- El color del fondo es demasiado oscuro o demasiado claro.
- Los colores del fondo no se integran bien con los de la escena.
- Los reflejos no se procesan bien.

Estos problemas tienen diferentes causas que requieren diferentes tratamientos. Dado que los dos últimos tienen que ver con el uso de mapas y colores, se discuten más extensamente en el libro de simulación de materiales. En el apartado siguiente resumo los principales aspectos a tener en cuenta por lo que respecta al primero, que está relacionado más directamente con el control de exposición y afecta a cualquier escena con independencia de los materiales que se utilicen.

### Fondos demasiado oscuros o claros

En la mayoría de los casos esto será debido a un problema de exposición similar al que ocurre en la vida real: si nuestra vista (o nuestra cámara) está adaptada a un interior oscuro, el exterior parece excesivamente brillante. Y vi-

ceversa. Lo mismo ocurre en las escenas virtuales: si se utiliza una imagen de bajo rango dinámico para el fondo, con una iluminación diurna, puede resultar invisible. Por tanto, lo primero que habrá que hacer será comprobar el control de exposición.

En algunos programas, como 3ds Max, una solución simple, que en muchos casos puede ser suficiente (a partir de la versión 2010 de 3ds Max), es desactivar la opción *Process background and environment maps*. Al hacer esto, el algoritmo de control de exposición no aplica la redistribución de intensidades al fondo.

Sin embargo, esto puede provocar otro tipo de problemas por lo que no hay que descartar otras opciones algo más complicadas como las siguientes:

1. en el mismo panel de control de exposición se puede desactivar la opción predeterminada *Physical scale* e introducir un valor manual adecuado a través de la opción *Unitless*.
2. También se puede intensificar la luminosidad del mapa de fondo editándolo y aumentando el valor de *Output amount*;
3. Se puede combinar el *shader Gamma & Gain* que he descrito más arriba con el mapa de fondo y asignar un valor alto al multiplicador de *Gain* para conseguir lo mismo. Pero son opciones a las que es mejor no tener que recurrir pues introducen artificios considerables en la imagen.

Si en lugar de un fondo virtual se utiliza un objeto, tal como un plano, para el fondo, los problemas anteriores desaparecen pues en este caso el objeto está integrado en la escena. Pero también puede ocurrir que no resulte visible. Sin embargo, en este caso la explicación es más simple y más fácil de evitar. Lo más probable es que no resulte visible por la sencilla razón de que no recibe luz, pues las luces estarán dirigidas hacia objetos de la escena. En este caso, para no alterar la iluminación general, lo más sencillo es iluminarlo frontalmente con una luz directa y utilizar el botón *Exclude*, de los parámetros de la luz, para excluir todos los objetos de la escena excepto este fondo.



Si se quiere comprobar exactamente lo que puede ocurrir en los casos principales, citados anteriormente, probar lo que sigue.

1. Crear una escena sencilla, un objeto sobre un plano, o abrir una escena que incluya un fondo. Iluminar la escena con un sistema de luz diurna *mr* (*mr sun* y *mr sky*).
2. Si se utiliza el fondo dado por defecto, *mr Physical sky* no habrá ningún problema pues este es un mapa procedural que se adapta a las características de la escena.
3. En lugar del *Physical sky* utilizar como fondo un mapa corriente (LDR) que represente, por ejemplo, un cielo.
4. Comprobar que:
  - i) Si se mantiene activado el procesamiento del fondo, con la exposición correcta para este caso (EV 15,0, más o menos), el fondo sale negro. Esto es debido a que el fondo no está iluminado con las mismas luces que el resto de la escena.
  - ii) Si se reduce la exposición al mínimo (EV 0,0 o poco más), el fondo se procesa correctamente pero la escena sale quemada.
  - iii) Si se mantiene la exposición correcta (EV 15,0) y se desactiva el procesamiento del fondo, la escena se procesa correctamente y el fondo también.

Las imágenes de la figura 4.39 ilustran estos tres casos. Queda por añadir que hay un caso, de gran importancia práctica, donde la solución anterior no sirve pues el fondo debe integrarse en la escena. Eso ocurre cuando utilizamos un material de tipo *matte/shadow* que se usa para fusionar un escenario virtual con una imagen de un escenario real. Este caso también se analiza y se ilustra en el libro sobre simulación de materiales.

→ 5



# Iluminación directa. Luces y Sombras

## Nota sobre los ejemplos

Este capítulo se concentra en el análisis de las técnicas principales de creación y configuración de luces. El análisis se basa exclusivamente en la luz directa y se prescinde de todo lo relativo a la iluminación indirecta, que requiere sistemas de cálculo avanzado y se desarrolla en el siguiente capítulo.

Para que la explicación sea sintética y fácil de seguir, los ejemplos se han reducido a lo esencial y la mayoría de ellos están basados en un cubo de 3x3x3 metros.

Para poder reproducir los ejemplos del modo más exacto posible, en 3ds Max, hacer lo siguiente (el procedimiento sería similar en otros programas):

1. Establecer las unidades en cm (menú *Customize/Units Setup*, unidades genéricas, botón *System Units Setup*).
2. Crear una *box*. Situarla en 0,0,0 y hacer que sus dimensiones sean de 300x300x300. Añadirle un modificador *Shell* con un valor de *Outer Amount* de 20. De este modo se da grosor a las superficies de la caja. Convertirla a malla editable.
3. Crear una cámara situada fuera de la caja, a 150 cm de altura, apuntando al centro del interior de la caja. Activar los planos de recorte para poder ver el interior, el *Near Clip* rebasando ligeramente el muro más cercano a la cámara y el *Far Clip* más allá del muro más lejano.
4. Con *mental ray* no se reconocen los planos de recorte si se utilizan sus algoritmos de *render* predeterminados. Como la mayoría de los ejemplos se desarrollarán con *mental ray* (si no estuviera activo este motor activarlo desde *Render Setup/Common/Assign Renderer*) hay que

ir a *Render Setup/Renderer/Rendering Algorithms* y marcar la casilla "Enable" del grupo *Scanline*. Por defecto este algoritmo está activado cuando se usa el motor predeterminado de Max y desactivado cuando se usa *mental ray*. Pero en este último caso hay que activarlo para que procese los planos de recorte.

5. En *Render Setup/Common*, elegir un tamaño adecuado para la salida de *render*. Asegurarse de que, en la parte inferior, está seleccionado el visor de salida que interese, que en general será el de cámara y presionar el candado que hay a su derecha para bloquear esta salida. Presionar el botón *Render* para activar el proceso y, si se prefiere, en los siguientes, el botón F9 para repetir el último o el botón *Render Production* de la Barra de herramientas, lo que será más cómodo pues nos evitará tener que volver a entrar en *Render Setup* mientras no cambiemos la configuración.
6. Seguir las indicaciones que se darán en cada ejemplo para completar las configuraciones de luces o especificaciones de salida.

Debe tenerse en cuenta que en algunos casos se comentarán diferencias bastante sutiles, que sólo pueden apreciarse adecuadamente en pantalla, una razón adicional para considerar los ejemplos que daré como una referencia que sólo será realmente útil si se recrea personalmente.





## 5.1 Luces simples puntuales, focales, directas

### Interfaz y métodos de creación y modificación

Todos los programas cuentan con recursos para crear este tipo de luces, que ya se han descrito de modo general en el capítulo 2, y que aquí se describirán desde el punto de vista técnico de su utilización en programas de aplicación concretos. El **interface** es muy simple y no es necesario recordar sino cosas muy elementales. En 3ds Max, por ejemplo, todo lo que hay que hacer es ir al panel *Create*, escoger un tipo de luz y pinchar en un punto de la escena para marcar la posición, en el caso de luces omnidireccionales o pinchar y arrastrar para marcar un punto y su centro de interés (*target*) en el caso de luces direccionales o focales. Si se utilizan luces direccionales o focales “libres” (sin *target*) la luz se crea en dirección perpendicular al plano en que se pincha.

En cualquiera de estos casos pueden utilizarse, después, los procedimientos corrientes de transformación para cambiar estas posiciones y los procedimientos específicos de modificación de propiedades desde el panel *Modify* para ajustar los parámetros que veremos a continuación.

En algunos programas, si no se asigna ninguna luz a la escena, se crean **luces predeterminadas**. En 3ds Max, a partir de la versión 4.0 se crean automáticamente una o dos luces, según la opción activa. Para cambiar esta opción ir al primer menú de Visor en entrar en *Viewport configuration* y escoger una o dos luces (si se escoge una, se crea una luz omni denominada “Luz clave predeterminada” (*Default Key Light*) que se sitúa frente a un objeto y que se creará en el centro de la escena y desplazada hacia arriba y hacia la izquierda; si se escogen dos, se crea una luz adicional, denominada “Luz de relleno predeterminada” (*Default Fill Light*), en posición opuesta a esta, iluminando la parte en som-

bra del objeto desde abajo y hacia la derecha. Como es obvio, es preferible crear luces adecuadas para la escena en lugar de depender de las predeterminadas. Al crear una luz, de cualquier tipo, desaparecen estas luces predeterminadas aunque en las últimas versiones de este programa hay que especificar, en la configuración del visor que se utilicen las luces que el usuario ha creado en lugar de las predeterminadas.

En general conviene seguir el mismo criterio y crear no una sino dos luces, una principal y otra secundaria, en posición opuesta. Más adelante veremos ejemplos de estos tipos de configuración.

Una vez creada una luz se pueden modificar sus propiedades, su posición y su orientación. Las **propiedades** las analizaremos en el siguiente apartado. Para **modificar su posición y orientación** seleccionarla y moverla como un objeto cualquiera. Las luces pueden contar con un centro de referencia (*target*) que facilita la dirección a la que apunta. El que una luz cuente o no con *target* es una opción que también puede activarse o desactivarse. Una vez que la luz esté situada en una posición más o menos estable puede ser más práctico desactivar el *target* para que haya menos objetos en la escena. Y a la inversa.

También conviene ocultar las luces, una vez creadas, para que no molesten (esto puede hacerse en 3ds Max con un atajo de teclado, Shift+L). No perder de vista que aunque una luz esté oculta sigue iluminando. Para apagarla hay que hacerlo expresamente: es una de las propiedades básicas que veremos a continuación.

En 3ds Max también se puede ajustar la posición de una luz focal o direccional haciendo que el visor muestre lo que ve la luz. Para ello, activar un visor y presionar la tecla «\$». O ir al menú de Visor/ *Vistas* y escoger la luz. Los controles de navegación cambian, adaptándose a los parámetros propios de la luz, lo que permite acercarla, desplazarla, cambiar los ángulos de cono en el caso de luces focales, etc. Pero lo que importa es que nos permite controlar el modo en que la luz



se sitúa con respecto a los objetos y que nos muestra los contornos de la sombra, pues los contornos que “ve” la luz coinciden con los contornos de la sombra.

Otro modo de mover una luz focal o direccional es reorientarla a partir de la posición de un punto de reflejo. Si, por ejemplo, queremos que un elemento curvo, muestre un reflejo en una determinada posición, lo que nos llevaría a seleccionar la luz e irla moviendo hasta que el reflejo caiga en ese punto, basta con ir al menú *Tools/ Align/ Place Highlight* o activar el icono correspondiente en el menú estándar, junto a los botones generales de alineación. A continuación, marcar el punto que interese sobre el objeto: la luz se moverá automáticamente para que el reflejo caiga sobre ese punto.

Pasaré por alto otra luz simple que ya está en desuso aunque se sigue encontrando en la mayoría de programas: la luz *Ambient*. A diferencia de las anteriores esta se ajusta a través de un menú especial (en 3ds Max, el menú *Rendering/ Environment/ Common Parameters*). La luz ambiente ilumina por igual todas las zonas en sombra para simular el efecto de interacción entre objetos, que hace que las partes en sombra de los objetos queden parcialmente iluminadas. Como esto altera el color y la intensidad de *todos* los objetos es preferible utilizar luces secundarias para iluminar las partes en sombra o, mejor todavía, utilizar sistemas de luz avanzada. Otra opción igualmente artificiosa pero que da menos problemas, sería asignar a algunos objetos valores de autoiluminación. Pero, como digo, lo más recomendable es utilizar luces secundarias para suavizar las sombras.

### Parámetros básicos de luces simples

En lo que sigue me referiré exclusivamente a los parámetros de 3ds Max para que la explicación sea más simple. Pero las propiedades básicas son similares en todos los programas.

Los parámetros **generales** permiten activar o desactivar la luz, cambiar el tipo de luz, y especificar el tipo de cálculo de sombras. Permiten por añadidura excluir algunos objetos.

Los parámetros de intensidad, color y atenuación permiten ajustar estos tres valores: La **intensidad** se controla con un multiplicador que utiliza unidades arbitrarias. Un valor de 1.0 quiere decir la intensidad será la “adecuada” para la calibración media de un monitor corriente. A partir de aquí se pueden

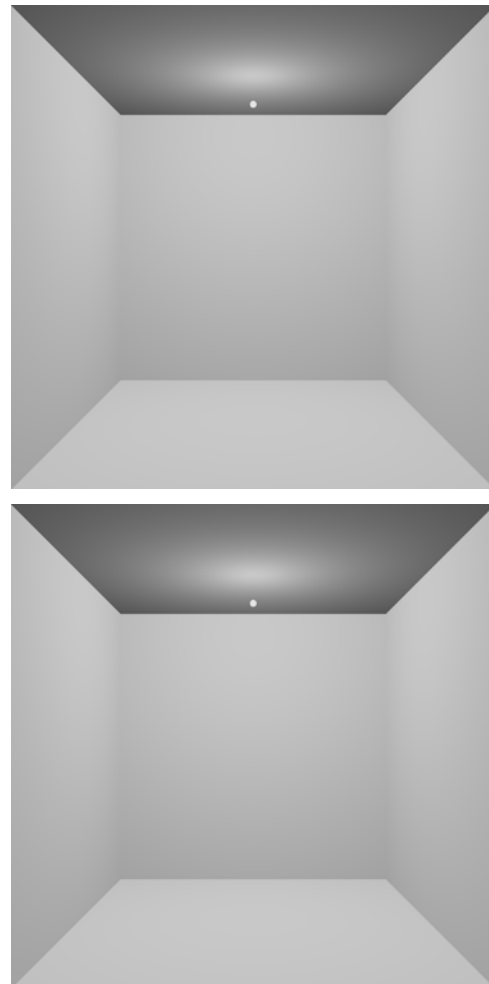


Figura 5.1 Luz omnidireccional situada simétricamente con respecto a los 6 planos de un recinto regular (a) y asimétricamente (b). En este último caso, el suelo resulta más iluminado que el techo, pese a estar más lejos, debido a que la iluminación sólo depende del ángulo.

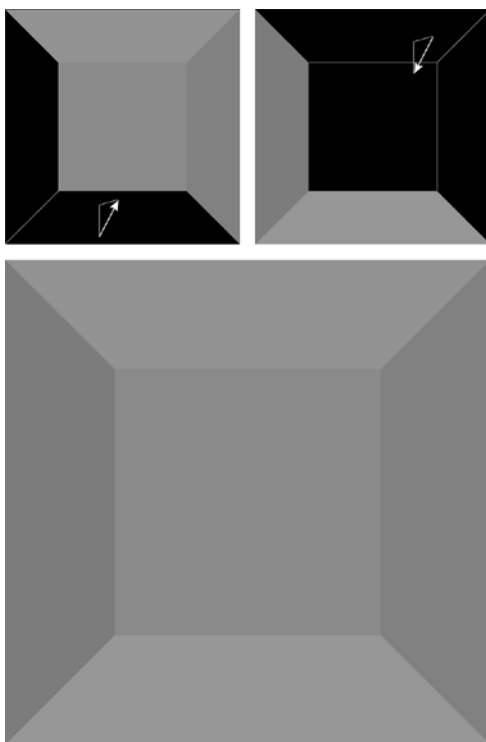


Figura 5.2 Luces direccionales simples dispuestas de modo que simulan la iluminación secundaria debida a los rebotes en las superficies de un recinto no iluminadas directamente

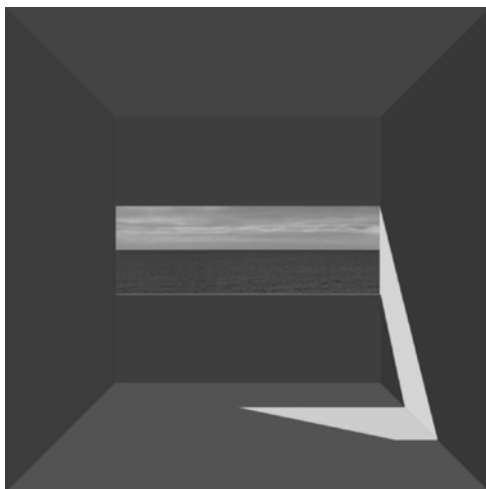


Figura 5.3 Luz direccional utilizada para simular el sol, combinada con las luces secundarias del ejemplo anterior

dar valores inferiores a 1.0 para atenuarla o superiores a 1.0 para intensificarla. El **color** se asigna por medio de un panel característico, un selector de colores con controles RGB o HSV. En general convendrá dejar el color predeterminado, blanco.

Las luces directas y focales tienen parámetros adicionales para controlar el ángulo de la **zona interna (haz, hotspot)** de intensidad uniforme y de la **zona externa (campo, falloff)** en donde la intensidad de la luz pasa de 1 a 0. También cuentan con parámetros para controlar la **forma de las áreas de luz del foco**: si se escoge la opción rectángulo se activan opciones que permiten modificar la proporción (en el caso de forma rectangular) o ajustarla a un mapa de bits particular en el caso de utilizar un proyector (en el grupo siguiente). Hay otros controles importantes como **overshoot (rebasar)**: si se activa, el foco emite luz en todas direcciones (como una omni) pero sólo calcula las sombras dentro de la zona de atenuación.

Hay otros parámetros, como los propios de control de las **sombras** que se explicarán en el apartado correspondiente, más adelante. Y otros, como los de volumen de luz y otros efectos que también veremos en otra sección más avanzada.

Cuando se está preparando la iluminación de una escena hay que hacer ajustes constantes en los parámetros principales, sobre todo en la intensidad de la luz. Es posible acceder de un modo más ágil a estos parámetros principales de cada tipo de luz, por medio de un cuadro de diálogo especial que, en 3ds Max, se abre desde el menú *Tools/Light Lister* y que presenta estos parámetros para todas las luces de la escena o para las luces seleccionadas (si se marca la opción correspondiente del panel).

Es importante tener bien presentes las limitaciones de las luces simples pues se basan exclusivamente, como ya he dicho, en el ángulo entre el rayo de luz y la normal a la superficie. Esto puede provocar efectos extraños, algunos de los cuales se ilustran en las figuras adjuntas.



Las luces omnidireccionales son muy fáciles de utilizar pero son muy poco recomendables, en general, pues crean halos en las superficies sobre las que inciden perpendicularmente. Aparte de que este efecto es, en sí mismo, poco deseable, los halos pueden ser más intensos en superficies alejadas que en superficies próximas debido a que el ángulo de incidencia es menor. Este efecto, que puede resultar desconcertante si no se entiende bien el principio que está actuando, se ilustra en la figura adjunta.

Las luces directas pueden utilizarse para simular la iluminación secundaria de las paredes. Si se disponen, como en las figuras adjuntas, con una luz dirigida hacia el techo y el muro derecho y otra dirigida hacia el suelo y el muro izquierdo, el efecto se aproxima al de un sistema de luz global que computa la interacción entre objetos pero con un tiempo de computación muy inferior.

Las luces directas también pueden utilizarse para simular la iluminación del sol. En otra de las figuras adjuntas a este apartado se ha creado una luz directa que entra por la ventana, con sombras de *ray trace*, y un fondo de intensidad adecuada a la escena que se quiere simular. Aunque el resultado no tiene los matices que se obtienen en un sistema de iluminación avanzada, el tiempo de computación es muy inferior y la simplicidad de la escena puede ser adecuada para ilustrar un proyecto en el que se prefiera esta simplicidad a la de una representación muy exacta.

Las luces focales (*spot*) se pueden utilizar para simular, obviamente, focos pero también luces extensas no muy grandes pues la apertura de la zona externa crea un ligero efecto de penumbra. Pero este efecto es limitado como puede apreciarse en la última figura adjunta de este apartado.

### Atenuación con luces simples

Aunque las luces simples no disminuyen de intensidad con la distancia se puede introducir un factor de **atenuación** mediante recursos complementarios que proporcionan un

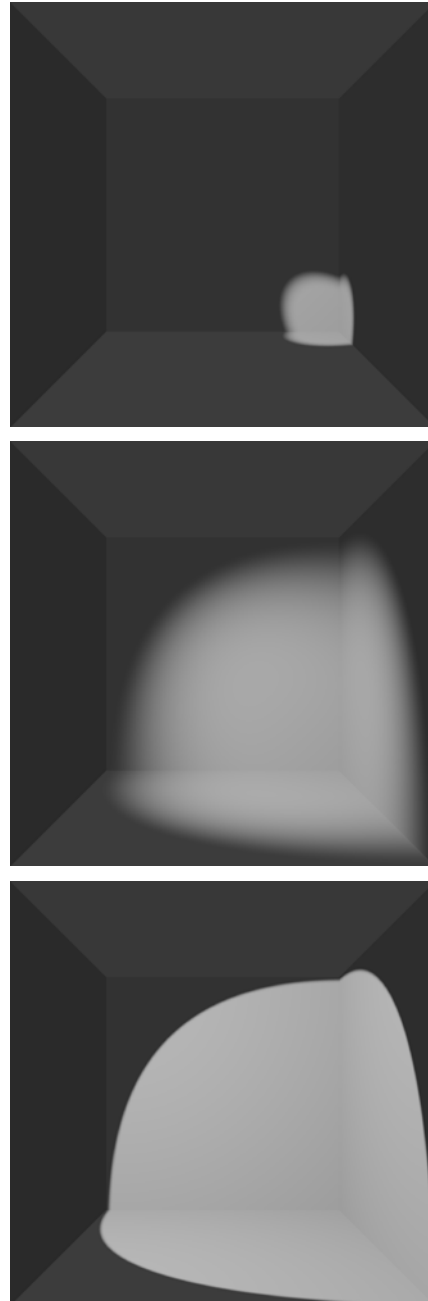


Figura 5.4 Luz spot (situada en la esquina superior izquierda frontal) con diferentes configuraciones de zona interna/externa (hotspot/falloff): a) 10°/15°, b) 10°/75°, c) 73/75°

grado de control bastante sofisticado y que puede substituir, incluso con ventaja, a los de las luces fotométricas. La mayoría de los programas cuentan con medios para especificar diversos tipos de atenuación y combinarlos entre sí.

El medio más simple es especificar un determinado tipo de atenuación que se denomina, según los programas y los tipos, *decay*, *attenuation* o *distance falloff*. La mayoría de

los programas permiten escoger entre cuatro tipos:

- |   |                                  |
|---|----------------------------------|
| 0 | Ninguno                          |
| 1 | Lineal (inversa)                 |
| 2 | Cuadrática (inversa al cuadrado) |
| 3 | Cúbica                           |

En otros casos es posible especificar un exponente con decimales: un valor tal como 1.5, por ejemplo, haría que la atenuación se produjera con una velocidad intermedia entre lineal (1) y cuadrática (2).

De los cuatro tipos principales citados, el primero (ninguna atenuación) es casi siempre el predeterminado. *Lineal* hace que la intensidad de la luz disminuya en proporción directa a la distancia. *Cuadrática* hace que la intensidad disminuya según el cuadrado de la distancia. Esta es la ley clásica descubierta por Lambert en el siglo XVIII y es la que se cumple en la realidad en condiciones ideales. Sin embargo, debido a que estas condiciones ideales no se dan en la práctica (debido a que hay un alto grado de iluminación ambiental y esta especificación, por sí sola, puede dar resultados demasiado bruscos) puede ser preferible escoger la primera. *Cúbica* (3) hace que la intensidad disminuya con mayor rapidez, según el cubo de la distancia. Es una opción que raramente se utiliza y ni siquiera está disponible en muchos programas. Puede ser útil para situaciones tales como una escena cubierta por una espesa niebla o dentro del agua o para simular la luz de una vela. Pero en estos casos se pueden utilizar también otros recursos más efectivos como los que se explican a continuación.

Otra alternativa es fijar distancias concretas a partir de las cuales comienza y termina la atenuación. Esto es notoriamente artificioso pero puede ser un recurso muy eficaz para controlar la luz. En 3ds Max, como en otros programas, se puede controlar la atenuación por medio de dos esferas concéntricas que representan los rangos de atenuación. La opción más corriente es la de *atenuación lejana* (*far attenuation*): entre *inicio* y *fin* la luz co-

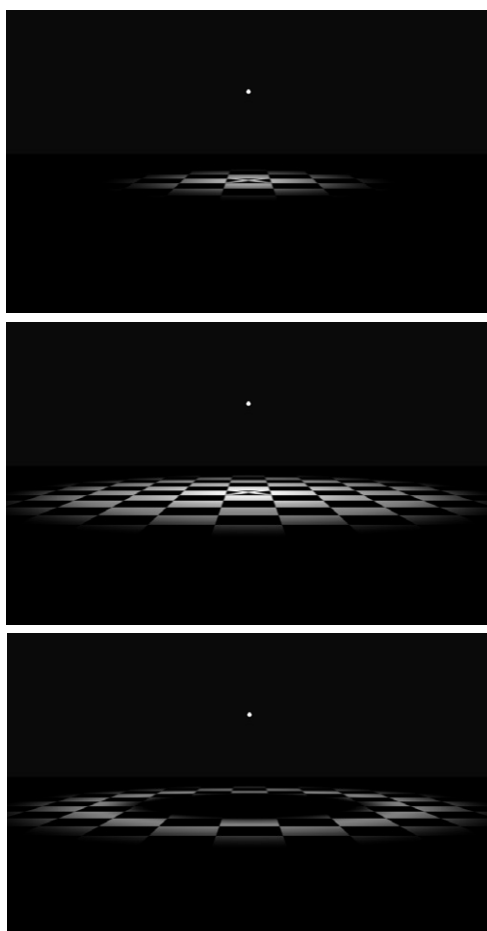


Figura 5.5 Atenuación con luces simples: luz omni colocada a 100 cm de altura sobre un suelo de baldosas (de 50x50 cm) marcada con un punto blanco: a) con atenuación lejana 50/200 (inicio/fin), b) con atenuación lejana 200/300, c) con atenuación cercana 150/200 y lejana 200/300



mienza a disminuir entre 1 y 0. La opción de *atenuación lejana* (*near*, un término equívoco) hace lo contrario: la luz aumenta entre *inicio* y *fin* hasta alcanzar el valor 1 en *fin*.

La figura adjunta muestra un par de ejemplos de los efectos que se pueden conseguir. Los dos primeros son casos corrientes en los que la luz comienza a disminuir a partir de una determinada distancia hasta que se extingue. El tercero es un ejemplo más sofisticado en el que la luz no comienza a actuar hasta una determinada distancia y luego decrece en el rango especificado, lo que crea un efecto de anillo en una zona de la escena. Es un recurso muy útil si, por ejemplo, se quiere simular el efecto de una luz colocada sobre un soporte que arrojaría sombras. El cálculo real de las sombras del soporte resultaría mucho más lento y podría dar resultados imperfectos.

### Recursos adicionales. Exclusión. Afectar a difuso. Afectar a especular

En los escenarios virtuales se pueden hacer con facilidad varias cosas que no son posibles en escenarios reales. Uno de estas cosas es la **exclusión**: utilizar luces que no afecten a determinados objetos. Este recurso, denominado también *exclusión selectiva* o *enlace selectivo* (*light linking*, *selective lighting*) es muy útil y se utiliza por varias razones, principalmente para ajustar la iluminación de un objeto o grupo de objetos que nos interesan particularmente (excluyendo todos los demás) o bien para iluminar un objeto que actúa como fondo sin afectar al resto de la escena. A menudo esto se lleva a cabo clonando una luz general que ilumina a varios objetos y haciendo que la nueva copia ilumine sólo a uno de los objetos que interesa, con lo que se asegura que la dirección de la copia es idéntica a la de la original. Hay programas, como *SoftImage* que utiliza este recurso de dos modos, *inclusive* y *exclusive*. Es un recurso muy efectivo pero que debe utilizarse con moderación pues puede alterar la iluminación de la escena creando efectos excesivamente artificiosos.

Otro recurso interesante es hacer que una luz afecte al color local pero no a los reflejos y viceversa. La reflexión difusa hace que los rayos se disperse en todas direcciones. La reflexión especular, que se concentran en una dirección. Si se utilizan luces estándar, la reflexión especular correspondiente a una luz que no es sino un punto en el espacio, sería invisible pues afectaría a un punto infinitesimal de la superficie. Una reflexión adecuada requeriría métodos tales como *ray trace*, que computan efectivamente la relación del objeto con otros objetos. Pero con luces estándar la luz sólo se relaciona con cada uno de los objetos de la escena de modo independiente.

Para simular el efecto de una iluminación especular, el *shader* aplicado al material crea una zona especular de tamaño ajustable. Pero dado que este efecto puede ser deseable en unos casos pero no en otros, muchos programas, tienen un control que permiten desactivar el efecto haciendo que una luz sólo afecte al componente difuso, sólo al componente especular o a ambos (la opción predeterminada). En 3ds Max basta con seleccionar la luz y abrir el menú contextual con el botón derecho del ratón y marcar o desmarcar las opciones *Affect Diffuse* o *Affect Specular*. En el libro de Simulación de Materiales, en el capítulo sobre reflejos, se proporcionan ejemplos concretos de utilización de este recurso para ajustar los reflejos.

### Luces simples extensas

Las luces puntuales tienen la importante ventaja, ya mencionada, de que son muy rápidas. Y la importante desventaja, entre otras, de que no crean efectos de umbra y penumbra como las luces reales que siempre emiten a través de una superficie con una determinada extensión.

Sin embargo, se pueden utilizar luces extensas en la mayoría de los programas modernos de simulación. En 3ds Max esto se hace por medio de luces especiales de *men-*





*tal ray* que se incluyen en la lista de luces estándar. Hay dos tipos, *mr Area Spot* y *mr Area Omni*. La primera crea luces planas de dos tipos: rectángulos y discos. Y la segunda crea luces de volumen de dos tipos: esferas y cilindros. Se describen en los dos siguientes apartados, referidas a 3ds Max pero son muy similares en otros programas.

### Luces extensas planas

Si se utilizan luces planas en 3ds Max (*mr Area Spot*), la diferencia con respecto a las luces simples es que aparece un nuevo grupo de parámetros, “parámetros de luz de área” (*area light parameters*) que permiten elegir entre dos tipos, disco y rectángulo y, en cada caso, las dimensiones correspondientes: radio, en el caso de disco y anchura y altura en el caso del rectángulo.

En sistemas de cálculo simple esto no tiene efecto apreciable en la distribución de la luz y sólo afectará a las sombras que mostrarán efectos de umbra y penumbra. Por tanto retomaremos este tema en el apartado correspondiente de la sección sobre sombras.

### Luces extensas de volumen

Si se utilizan luces de volumen en Max (*mr Area Omni*), la diferencia con respecto a las luces simples es que aparece un nuevo grupo de parámetros, “parámetros de luz de área” (*area light parameters*) que permite elegir entre dos tipos, esfera y cilindro y, en cada caso, las dimensiones correspondientes: radio, en el caso de esfera y radio y altura en el caso de cilindro.

En sistemas de cálculo simple esto tampoco tiene efecto apreciable en la distribución de la luz y sólo afectará a las sombras que mostrarán efectos de umbra y penumbra. Por tanto también retomaremos este tema en el apartado correspondiente de la sección sobre sombras.

## 5.2 Luces fotométricas

Las luces fotométricas simulan de un modo más exacto las luces reales y se utilizan con sistemas de iluminación global. A diferencia de las anteriores, utilizan unidades fotométricas con significado físico correcto, su intensidad disminuye con la distancia y siempre cuentan con recursos adecuados para simular la extensión.

Se pueden utilizar tres tipos básicos: puntual, lineal y de área, aunque los parámetros principales son los mismos: Cada uno de estos tipos puede contar con accesorios tales como un *target* que facilite su reorientación y que se puede activar o desactivar, como con las luces simples.

Además de estos tres tipos básicos hay sistemas fotométricos de luz diurna que se verán en el siguiente apartado.

### Parámetros básicos de luces fotométricas

Los parámetros básicos de las luces fotométricas son similares a los de las luces estándar. Al igual que en éstas, hay un grupo de parámetros generales para activar o desactivar la luz, activar o desactivar el *target* para facilitar la orientación, escoger el tipo de sombras, excluir o incluir objetos, una sección para ajustar los parámetros de sombras, etc.

Una de las diferencias, desde el punto de vista de los parámetros básicos, es que, en principio (con la excepción que veremos), no hay un grupo de parámetros para ajustar la zona interna y externa del foco, pues estos son artificios que no se requieren dado que estas luces simulan distribuciones reales.

Tampoco es necesario ajustar la atenuación pues estas luces se comportan como las luces reales y disminuyen de intensidad con la distancia. Pero sí se suelen incluir parámetros para ajustar la atenuación lejana, con un valor que especifica el inicio (*start*) y otro que especifica el final (*end*) de la iluminación. Esto puede utilizarse para crear efectos artificiales en la escena. Pero esto no modifica



el tipo de atenuación que siempre disminuye con el cuadrado de la distancia.

También incluyen parámetros para especificar la forma y las dimensiones. En las últimas versiones de 3ds Max se incluyen los siguientes tipos:

1. *Point*. La luz se emite desde un punto sin dimensiones.
2. *Line*. La luz se emite desde una línea y el panel cambia para mostrar el parámetro de longitud.
3. *Rectangle*. La luz se emite desde un área rectangular y se muestran dos parámetros para ajustar estas dimensiones.
4. *Disc*. La luz se emite desde un área circular y el panel cambia para mostrar el parámetro de radio del círculo.
5. *Sphere*. La luz se emite desde un volumen esférico y el panel muestra el parámetro de radio de la esfera.
6. *Cylinder*. La luz se emite desde un volumen cilíndrico y el panel muestra el diámetro y la altura del cilindro.

Como la forma de la luz afecta más claramente a las sombras también retomaremos este tema en la sección correspondiente sobre sombras.

Además de estos parámetros básicos, las luces fotométricas incluyen dos grupos de parámetros adicionales que marcan la diferencia principal con las luces simples: distintos tipos de unidades y distintos tipos de distribución.

### Unidades de luces fotométricas

Es importante recordar que las unidades con que se trabaja en el modelo deben ser coherentes para que las luces fotométricas funcionen adecuadamente. Hay bastantes usuarios que se desprecupan por cómo se registran internamente las unidades pues, *para ellos*, “1” significa “1 metro”. Pero si las unidades han quedado registradas internamente en cm, pulgadas, metros o pies, el resultado de aplicar una luz con una intensidad determinada será completamente distinto.

Para cambiar las unidades generales, en 3ds Max, ir al Menú *Customize/ Units Setup* y comprobar todo lo que sigue: a) En *System Units Setup*, hacer que en el grupo *System Units Scale* especifique que 1 unidad = 1.0 = Centímetros. O milímetros o metros. Pero recordar que debe evitarse que 3ds Max trabaje con números inferiores a 1.0 o demasiado grandes, porque la navegación se dificulta y hay saltos bruscos o errores aparentes; b) Comprobar que en el Grupo *Lighting Units* está marcada la opción *International*; c) Comprobar que la Escala de Unidades de presentación, *Display Units Scale*, es *Generic Units*. Las demás opciones que aparecen en este panel se utilizan para traducir unidades. Es decir que, si por alguna razón hemos recibido un archivo cuyas unidades están en pulgadas y no queremos reescalarlo todo (algo nada recomendable en cualquier caso) podemos marcar la opción “Metric” y escoger, metros (o centímetros o lo que sea) para que el programa traduzca las unidades de pulgadas a metros. En este caso el programa colocará una inicial “m” (o “cm” o lo que sea) al final de cada cifra, lo que nos recordará que estamos utilizando unidades traducidas. Dado que algunos cálculos de la aplicación se basan en las unidades del sistema es recomendable no abusar de este recurso y utilizar siempre que se pueda unidades genéricas basadas en unidades internas del sistema.

Una vez que hemos comprobado que las unidades generales son coherentes, para ajustar las intensidades de la luz ir a la sección correspondiente y escoger una de las tres disponibles. Si no se recuerda bien el sentido de estas unidades revisar las explicaciones dadas en el primer capítulo.

a) Escoger **lm** si se quiere especificar la intensidad de la fuente en **lúmenes**. Para relacionar un valor en lúmenes con un tipo de lámpara cuya intensidad se da en vatios se puede utilizar una tabla como la siguiente que relaciona tipos de lámpara con rendimientos en lúmenes por watio:

Incandescente	10-17 lm/w
Halógena	12-36
LED	30-60
Fluorescente	35-100
Mercurio	50-140
Sodio	60-180
Máximo teórico	225

Es decir, que si utilizamos una lámpara halógena de 100 vatios con un rendimiento de 20 lm/w la intensidad nominal de partida será de 2.000 lm.

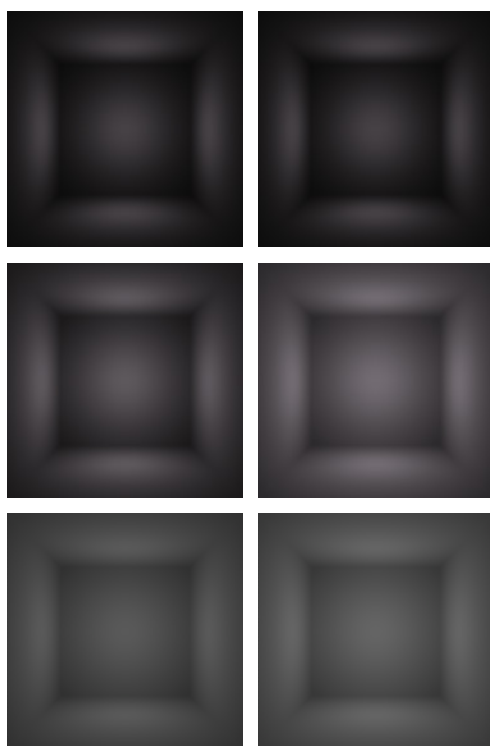


Figura 5.6 Variantes de iluminación de un interior combinando diferentes valores de intensidad y control de exposición:

- a) Luz omni standard
- a) Luz omni standard con intensidad 1.0
- b) Luz fotométrica, 1500 cd, sin FG, sin CtrlExp
- c) Id. Con FG, sin rebotes, sin CtrlExp
- d) Id. Con FG, dos rebotes, sin CtrlExp
- e) Id. Con FG, dos rebotes, 150 cd, EV7
- f) Id. Con FG, dos rebotes, 1500 cd, EV10

b) Escoger **cd** si se quiere especificar la intensidad de la fuente en **candelas**. Como los lúmenes se distribuyen sobre una esfera de 1 m de radio, y la superficie de la esfera es  $4\pi r^2$  para hallar la intensidad en candelas habrá que dividir los valores en lúmenes por  $4\pi$  (12.566). Y para pasar de candelas a lúmenes habrá que multiplicar por  $4\pi$  (12.566). El valor que se da por defecto en 3ds Max es 1500 cd lo que correspondería a 18.850 lúmenes, un valor muy alto que equivaldría a una lámpara halógena de alto rendimiento y más de 500 vatios. Pero es de suponer que se da este valor para que la escena sea visible si sólo se utiliza una luz.

c) Escoger **lx at** si se quiere trabajar con **luxes** a una determinada distancia. En este caso hay que especificar las unidades pero también la distancia a la que se quiere especificar esta iluminación, pues los luxes son una medida de la intensidad que recibe una superficie, no de la intensidad que emite una fuente. Al escoger esta opción, el valor predeterminado será el mismo que el dado en candelas para una distancia de 1 m. Es decir que si se ha especificado una intensidad de 1500 cd, al pasar a luxes nos encontraremos con que esto se traduce en 1500 luxes a 100 cm. Pero al escoger esta opción aparece una esfera virtual en torno a la luz que nos permite visualizar el campo que abarca esta especificación. Si aumentamos el valor del radio, aumentará esta esfera y podremos modificar el valor. Por ejemplo, si tenemos una luz situada a 3 m de altura y queremos que llegue a una mesa situada a 1 m de altura, con un valor de 500 lx, especificaremos “500 lx at 200 cm”. Si ahora volvemos a escoger cd nos encontraremos con que el valor en el origen será 2000 cd.

Resumiendo: se puede especificar la intensidad desde la fuente o desde el destino. En el primer caso la especificación es equivalente a escoger una luz con una determinada intensidad, dada globalmente, en lúmenes, o como densidad de intensidad, en candelas. En el primer caso la especificación es equi-



valente a escoger la iluminación con que queremos contar y dejar que el programa decida cual es la intensidad necesaria en el origen. Y todo esto depende, por otro lado, del tipo de distribución, que se explica más abajo.

## Unidades y control de exposición

En el capítulo anterior hemos visto que cuando se utilizan unidades reales es necesario ajustar el control de exposición a las condiciones de la escena. Nos encontramos entonces con que podemos conseguir diferentes efectos modificando los parámetros del control de exposición a partir de unos criterios que contendrán inevitablemente un alto grado de subjetividad, de apreciación personal y de relación con las intenciones del proyecto a que vaya destinada la escena.

En el caso de un exterior los casos generales están mejor estudiados y, como veremos, las variaciones son menores.

En el caso de un interior, el margen es mayor pues, en definitiva, estamos intentando replicar como se adaptaría el sistema visual a condiciones muy variables, en las que no puede determinarse de modo objetivo cuál sería el control de exposición adecuado para una determinada escena.

Sin embargo, para que esta elección sea lo más racional posible conviene comenzar a analizarla para casos muy simples.

Las figuras adjuntas muestran diferentes ajustes para un mismo interior, de 3x3x3 metros. La primera figura muestra los resultados utilizando unidades arbitrarias pero normalizadas: una luz omni, de intensidad 1.0, situada en el centro del espacio. Las siguientes figuras muestran los resultados correspondientes a una luz fotométrica, en la misma posición, con la intensidad dada por defecto (en 3ds Max) de 1500 cd y, en las siguientes, variando la intensidad y el control de exposición (en todos los casos se utiliza el control de exposición de *mental ray*, *mr photographic exposure control*, que permite ajustes más precisos). Como puede apreciarse en la segunda figura, los resultados utilizando una luz

fotométrica de 1500 cd, sin FG ni control de exposición, son prácticamente idénticos a los de una luz omni de intensidad convencional 1.0.

Si introducimos el control de exposición con los valores predeterminados para un interior (2.0) el resultado quedará quemado pues el recinto es muy pequeño, aunque 1500 cd no sea un valor demasiado alto. Así que tenemos dos soluciones para llegar a un resultado similar al de las figuras previas: aumentar el control de exposición o reducir la intensidad, como se ve en las siguientes figuras.

## Temperatura de color. Plantillas

Por defecto, el color predeterminado para luces fotométricas es el blanco normalizado D65 (correspondiente al cielo despejado con una temperatura de color de 6.500°K, como ya hemos visto en el Capítulo 1). Puede especificarse la temperatura de color escribiendo un valor determinado o escogiendo una plantilla. La primera sección de luces fotométricas en 3ds Max presenta una lista con tipos más o menos corrientes: incandescentes de 40 a 100 vatios, halógenas de potencia similar, fluorescentes, urbanas, etc. Al escoger uno de estos tipos predeterminados se activa automáticamente una determinada intensidad y una temperatura de color.

Si no hay razones muy claras para escoger una luz más cálida (por debajo de unos 5.000 o 4.000°K) o más fría (por encima de 7.000°K) lo mejor es dejar el valor predeterminado de 6.500°K.

## Tipos de distribución. Archivos IES. Otros archivos

Las luces fotométricas incorporan diferentes modos de distribución. En 3ds Max hay cuatro tipos de distribución disponibles que son similares a las que se encuentran en otros programas:

a) *Uniform Spherical*. La luz se distribuye uniformemente en todas direcciones, como una luz simple omnidireccional. Si se escoge

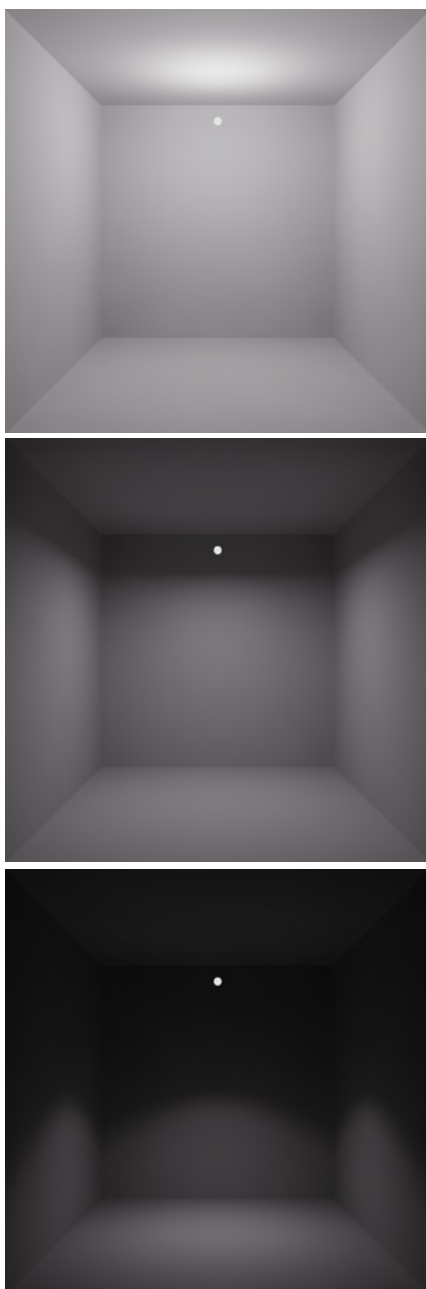


Figura 5.7 Luz fotométrica con las tres distribuciones básicas: a) uniforme esférica, b) uniforme difusa, c) spot con 90°/65°. Con spot 180°/120° los resultados serían los mismos que con (b) uniforme difusa

como forma de la luz una esfera o un cilindro esta es la única distribución disponible.

b) *Uniform Diffuse*. La luz se distribuye uniformemente en todas las direcciones del hemisferio a que apunta la luz.

c) *Spot*. La luz se distribuye uniformemente en el interior de un cono que cuenta, como las luces spot simples, con una *zona interna* (haz, *beam angle*) de intensidad uniforme y una *zona externa* (campo, *field angle*) en donde la intensidad pasa de 1 a 0. Tener en cuenta que si se utiliza *mental ray* el comportamiento es distinto que con luces simples debido a que la luz se atenúa siempre con la distancia: la intensidad decrece desde el centro (100%) hasta el borde extremo del campo central (*beam angle*), en donde se reduce a un 50% y luego sigue decreciendo hasta cerca del 0% en el campo total (*field angle*) aunque algo más de luz puede rebasar esta zona. También cuentan con parámetros para controlar la forma de las áreas de luz del foco pero en este caso están situadas en la sección de formas de la luz, que ya hemos visto que es común a todas las luces fotométricas.

d) *Photometric Web*. La luz se distribuye según un diagrama goniométrico que está incorporado a un archivo. En este caso se activa una sección nueva en la que aparece un botón <Choose Photometric Web> que da acceso al explorador del sistema para escoger un archivo adecuado.

Las tres primeras opciones son sencillas y se ilustran en las figuras adjuntas. La única diferencia con respecto a las luces simples es que el resultado es más real pues la intensidad se atenúa con la distancia. Tampoco hay diferencias apreciables entre *Uniform Diffuse* y *Spot* si esta última se configura con una apertura máxima de 180° (179.5 en 3ds Max) de la zona externa (*field angle*) y algo menor de la interna (*beam*).

La cuarta opción es la más interesante pues permite simular con exactitud el comportamiento de una luminaria real. Quizás convenga revisar, antes de continuar, los apartados correspondientes sobre este tema en el capítulo 1 (1.3, Tipos de luces).



IESNA91
[TEST] Test unknown
[MANUFAC] Manufacturing company unknown
TILT=NONE
1 74.11 1
7 1
1
2
0 0 0
1 1 0
0.00 8.00 11.00 15.00 20.00 30.00 90.00
0.00
300.00 220.00 180.00 130.00 75.00 0.00 0.00

IESNA91
[TEST] Test unknown
[MANUFAC] Manufacturing company unknown
TILT=NONE
1 190.46 1
6 1
1
2
0 0 0
1 1 0
0.00 13.00 25.00 38.00 55.00 90.00
0.00
200.00 160.00 110.00 65.00 0.00 0.00

IESNA91
[TEST] Test unknown
[MANUFAC] Manufacturing company unknown
TILT=NONE
1 304.31 1
4 1
1
2
0 0 0
1 1 0
0.00 30.00 60.00 90.00
0.00
100.00 85.00 50.00 0.00

Figura 5.8 Tres IES genéricos:

- a) Focal1.ies. Distribución focalizada;  
 b) Focal2SemiDistrb.ies. Distribución semifocalizada;  
 c) Focal3Distrb.ies. Distribución focalizada de modo relativamente uniforme

Para utilizar esta cuarta opción se necesita contar previamente con un archivo que especifique la distribución. Estos archivos pueden conseguirse de catálogos comerciales o se pueden crear manualmente. El formato más corriente es **IES** (*Illuminating Engineering Society*) aunque también pueden utilizarse los formatos LTI o CIBSE o Eulumdat. El formato **LTI** (creado por el *Danish Illuminating Laboratory*) se utiliza en los países escandinavos. El formato **CIBSE** (*Chartered Institution of Building Services Engineers*) se utiliza en el Reino Unido). El formato **Eulumdat** (extensión LDT) corresponde al formato europeo para lúmenes, equivalente a IES, se puede leer como un archivo de texto normal, y contiene información sobre el nombre del productor, el modelo de la lámpara, las dimensiones, la temperatura, el flujo luminoso, la potencia y la distribución de la luz. Pero el formato más utilizado en programas de simulación de la iluminación sigue siendo, por ahora, el IES.

El formato IES, desarrollado en Estados Unidos, formula la distribución de la intensidad luminosa (*luminous intensity distribution*, LID) en los nodos de una red, a partir de una serie de ángulos fijos en un plano horizontal y vertical de modo que se correspondan con un diagrama goniómetro (tipo C, uno de los más corrientes). Los datos están grabados en un fichero en formato ASCII con un determinado protocolo. La distribución de la intensidad viene dada por tanto por un diagrama en 3D que determina la intensidad en una determinada dirección.

El formato IES se basa en la siguiente estructura. En principio tiene que haber 16 líneas, aunque esto varía considerablemente en la práctica real y en los ficheros disponibles, con la siguiente información:

- 1) IESNA91 Referencia de la Illuminating Engineering Society of North America
- 2) [TEST] Referencia
- 3) [MANUFAC] Fabricante
- 4) TILT=NONE
- 5) 1
- 6) Número de lúmenes o “-1” si se utilizan cálculos absolutos

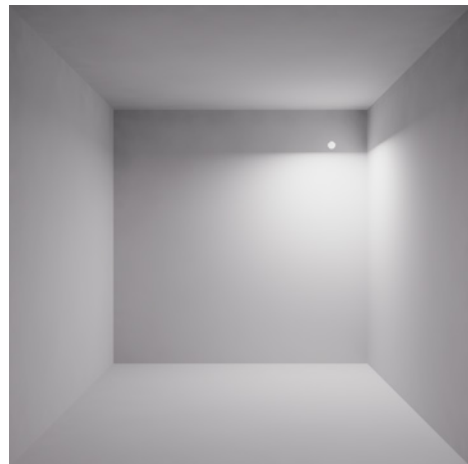
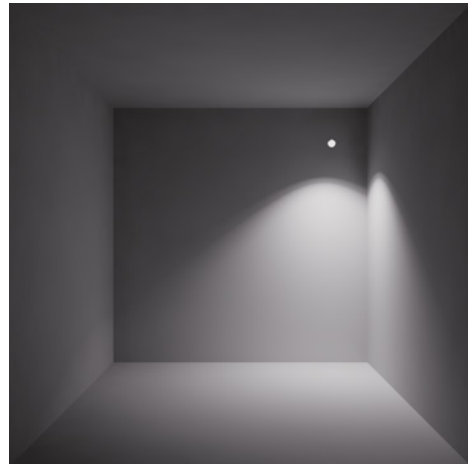
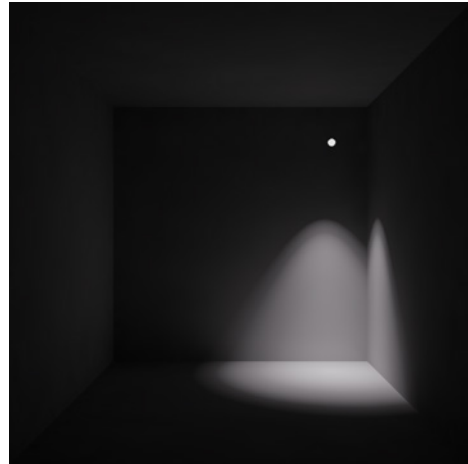




- 7) Multiplicador de los valores de intensidad en cd. Normalmente es "1" pero puede variarse el factor para reescalar las intensidades con facilidad, por ejemplo, para adaptarlos a otras unidades.
- 8) Número de ángulos verticales
- 9) Número de ángulos horizontales
- 10) 1
- 11) Tipo de unidad utilizado para medir la apertura luminosa: "1" indica pies, "2" indica metros
- 12) Anchura, longitud y altura de la apertura luminosa. Por lo general se definen como "0.0 0.0 0.0" lo que equivale a ignorarlas.
- 13) 1.0 1.0 0.0
- 14) Ángulos verticales, de menor a mayor. Si la distribución sólo apunta hacia abajo (hemisferio inferior) el primero y el último deben ser  $0^\circ$  y  $90^\circ$  respectivamente. Si apuntara hacia arriba deberán ser  $90^\circ$  y  $180^\circ$ . Y si cubren toda la esfera,  $0^\circ$  y  $180^\circ$ .
- 15) Ángulos horizontales, de menor a mayor. El primero debe ser  $0^\circ$ . El último indica el grado de simetría lateral: si es  $0^\circ$  la distribución es simétrica axialmente. Si es  $90^\circ$  es simétrica en cada cuadrante. Si es  $180^\circ$  es simétrica según un plano vertical. Si está comprendida entre  $180^\circ$  y  $360^\circ$  no hay simetrías laterales.
- 16) Valores de intensidad en cd. En primer lugar los valores correspondientes al primer ángulo horizontal, comenzando por el valor vertical más bajo y siguiendo por los valores verticales ascendentes en ese plano. A continuación los valores correspondientes al segundo ángulo horizontal y así hasta el último ángulo horizontal y sus correspondientes valores verticales. Cada conjunto de valores correspondientes a un plano vertical debe comenzar en una nueva línea.

Aunque se podría crear un archivo IES manualmente siguiendo esta estructura, los resultados no serían muy satisfactorios. Para crear un archivo IES manualmente se necesita contar con un programa adecuado en el que pueda especificarse la distribución y luego grabar los resultados en formato IES. El mejor programa para hacer esto, aunque es posible que ya no se distribuya, es *Lightscape*. El pro-

Figura 5.9 Distribuciones dadas por los tres IES anteriores, con una luz fotométrica situada en una esquina (a 50 cm de los planos adyacentes) con 8.000 cd de intensidad: a) *Focal1*, b) *Focal2SemiDistrb*, c) *Focal3Distrb*





cedimiento es sencillo y se explica en la ayuda de este programa. Hay un módulo específico en donde se puede crear interactivamente una red en 3D con la distribución de intensidades que se necesite para las direcciones principales. El programa se encarga de computar los valores intermedios para que el resultado sea continuo, cosa que no ocurriría si se introducen los valores a mano.

Con este programa, *Lightscape*, y la ayuda de Quim Moya, que trabajaba entonces conmigo como becario, creamos en 2007 tres ficheros genéricos cuyo contenido se da en las figura adjuntas. Como se ve, el resultado es bastante simple y el programa se encarga de escribir los valores automáticamente. Las líneas principales son las finales, en donde se especifican los ángulos y las intensidades respectivas. Quien quiera utilizarlos puede copiar su contenido en el Bloc de Notas de Windows y luego grabar el resultado con la extensión .IES. Una vez que estén situados en una carpeta del disco duro todo lo que hay que hacer es:

- a) Seleccionar la opción *PhotometricWeb* como tipo de distribución,
- b) En la nueva sección que aparecerá, *Distribution (Photometric Web)*, presionar el botón *<Choose Photometric File>*. Con esto se entra en el navegador de Windows para seleccionar el fichero.
- c) Una vez cargado el fichero aparecerá un pequeño diagrama con su distribución. Si es necesario, ajustar los parámetros *X Rotation*, *Y Rotation*, *Z Rotation* para cambiar su orientación. O hacerlo manualmente.
- d) Ajustar la intensidad. La intensidad predefinida de los tres archivos dados es muy baja (300, 200 y 100 respectivamente). Puede editarse y cambiarse a gusto del usuario.

Las figuras adjuntas muestran resultados característicos que se obtienen con estos archivos.

Para que el efecto sea más convincente se necesita, como en todos los casos anteriores,

asociar estas luces a luminarias. En el apartado sobre luminarias y objetos luminosos se completará este procedimiento de simulación.

Por último, recordar que muchos fabricantes de luces dan especificaciones de sus modelos en formato IES. Puede accederse con facilidad a sus páginas web buscando por las palabras clave "ies" y "nombreFabricante".

También es posible visualizar los archivos IES por medio de visores especiales (*IES Viewers*) que pueden descargarse gratuitamente por Internet. Hay muchos visores de este tipo. El que muchos usamos pues funciona muy bien es el creado por Andrey Legotin que puede descargarse (en 2011) de la página web de CG Arena, <http://www.cgarena.com/freestuff/tutorials/max/ieslights/>

### 5.3 Sombras

Los principios teóricos de los tres sistemas principales de cálculo de sombras ya se han visto en el capítulo 2. En esta sección me limitaré a comentar como se utilizan en la práctica, así como sus ventajas y desventajas que, de un modo muy sintético, puede decirse que se reducen a lo siguiente: en las versiones recientes, los mejores resultados se consiguen con sombras de *ray trace* asociadas con luces extensas de área. Pero el tiempo de cálculo es mayor y conviene conocer otras alternativas clásicas.

#### Parámetros generales

En 3ds Max, como en la mayoría de los programas de simulación, al editar una luz simple, hay tres secciones con parámetros que afectan al cálculo de sombras. En la primera, *General*, todo lo que hay que hacer es escoger el tipo de cálculo (*shadow map* o *ray trace*, con otras 3 variantes que carecen de interés). La segunda incluye algunos parámetros antiguos que están más bien obsoletos. Y la tercera varía con el tipo de sombra escogido.

En el caso de luces fotométricas tan sólo cuentan la primera y la tercera.



Esta segunda sección, que se encuentra en los parámetros de luces simples, *Shadow Parameters*, incluye dos grupos, *Object Shadows* y *Atmosphere Shadows*. El grupo *Object Shadows* incluye parámetros para variar el *Color* de las sombras (negro por defecto), su *Densidad* (su oscuridad) y para asignar un *Mapa* a la sombra que se funde con su color. Son parámetros artificiosos que se utilizaban para simular la iluminación avanzada o para crear efectos especiales (como hacer la sombra más luminosa haciendo el valor de *Density* inferior a 1.0, el valor predeterminado). El grupo *Atmosphere Shadows* se puede utilizar en combinación con efectos atmosféricos tales como Volumen de Niebla (*Volume Fog*). Si se activa, los efectos atmosféricos arrojan sombras en caso de que la luz se cruce en su camino; *Opacidad* (100%): permite hacer que las sombras sean más o menos transparentes; *Cantidad de color* (100%): ajusta el porcentaje en que el color de la atmósfera se funde con el de la sombra.

La tercera sección, principal, y común a las luces simples o fotométricas puede ser de dos tipos principales *Shadow Map Parameters* o *Ray Traced Shadow Parameters*, según el sistema de cálculo que se haya escogido en la primera sección, *General*. Las describo con más detalle en los siguientes apartados.

Por otro lado, si se utiliza *mental ray*, de los 5 tipos de sombras que incluye 3ds Max, *mental ray* sólo reconoce tres: *mr shadow map*, *shadow map* y *ray trace* (si se utilizan los dos que no reconoce, *mr* los procesa como *ray trace* y da un aviso). De hecho pueden reducirse a dos pues *shadow map* y *mr shadow map* son iguales con la ventaja a favor del último de que reconoce transparencias, puede incluir información de color y es preferible en ciertos casos (como cuando se utilizan los modificadores *hair and fur* o *motion blur*).

## Tipos de sombras

En todos los programas de simulación, hay dos sistemas básicos de obtención de sombras: *Mapa de sombras (shadow map)* y

### *Sombras de Raytrace.*

En 3ds Max, a partir de la versión 4.0 se utilizan otros dos más: *Sombras de área (Area shadow)* y *Raytrace avanzado (Advanced Ray-trace)*, que antes se obtenía como freeware de Blurbeta). Estos dos últimos tipos no añaden nada importante, en mi opinión, a los métodos principales por lo que no los tendré en cuenta en lo que sigue.

**Shadow map** (mapa de sombras) corresponde al sistema general que ya se ha descrito en la parte teórica, en el capítulo 2, y que captura la escena desde la posición del punto de luz, con lo que se obtienen los contornos que separan las zonas de luz de las de sombra. Estos contornos son reproyectados sobre la escena desde el punto de vista del observador.

**Raytrace Shadows** (sombras por trazado de rayos), también se ha descrito en aquél capítulo y se basa en el envío de rayos trazadores desde la posición de la luz que rastrean la escena calculando las intersecciones con los objetos. Su resultado es más exacto y producen sombras de contornos nítidos lo que puede ser deseable o no, según los casos. En general, se utiliza este método para sombras externas en condiciones de iluminación intensa. Las ventajas y desventajas son más o menos las inversas que el anterior. Utiliza menos memoria y reconoce objetos con transparencia y mapas de opacidad. Sólo se procesa una vez, excepto para objetos animados. Pero el tiempo de cálculo es mayor y no permite obtener sombras de bordes suaves.

## Sombras por mapa (*shadow maps*)

Las ventajas principales de este tipo son que permite obtener sombras de bordes más o menos suaves, que sólo se procesan una vez (excepto para objetos animados) y que es más rápido. Las desventajas son, que utiliza más memoria ( $4 \times n \times n$  bytes siendo  $n$  el tamaño en píxeles: un tamaño de 4096 píxeles supondría  $4096 \times 4096 \times 4$  bytes o 64 Mb) y, sobre todo, que no reconoce objetos con transparencia o mapas de opacidad. El que utilice

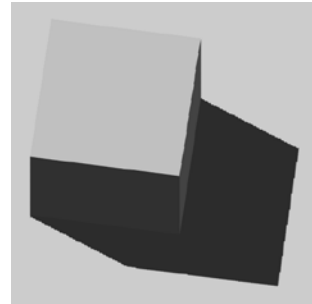
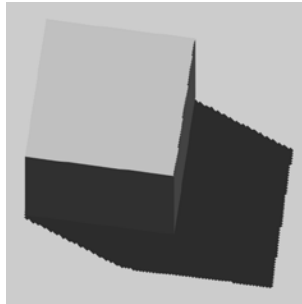
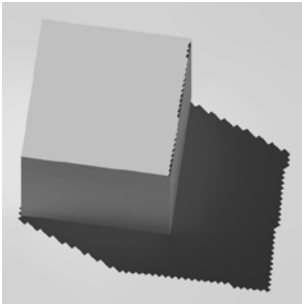


Figura 5.10 Sombras por mapa (mr). Variación del Tamaño (con sample range 0.01 y samples 4): a) 512, b) 1024, c) 2048

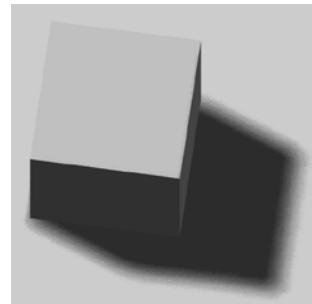
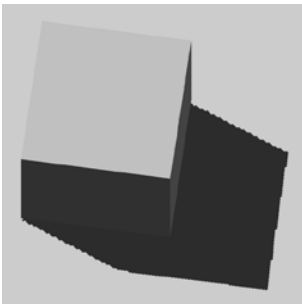


Figura 5.11 Sombras por mapa (de mr). Variación del Sample Range (con tamaño 1024 y samples 4): a) 0.0, b) 0.5, c) 1.5

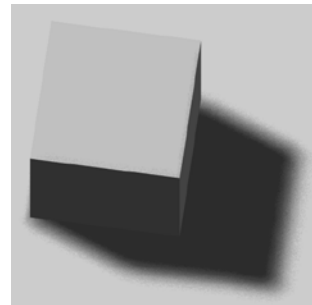
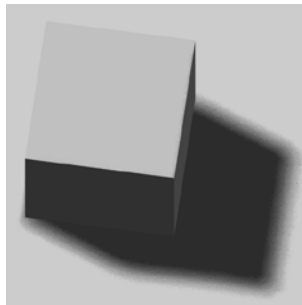


Figura 5.12 Sombras por mapa (mr). Variación del número de muestras (con tamaño 1024 y sample range 1.5): a) 1, b) 2, c) 4

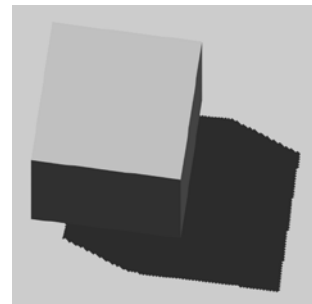
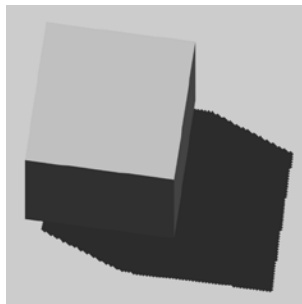
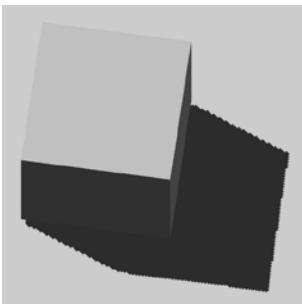


Figura 5.13 Sombras por mapa (mr). Variación del Bias (con tamaño 1024, sample range 0.01, samples 4): a) 0.5, b) 2, c) 4

más memoria es poco relevante con la potencia de los ordenadores actuales.

Las sombras por mapa *del motor predeterminado de Max* utilizan tres parámetros:

a) El **tamaño del mapa** (*Size*). Cuanto mayor sea la resolución con que se ha captado la escena mayor será la precisión del resultado y viceversa. Por otro lado, cuanto mayor sea esta resolución, mayor será el coste de memoria y el tiempo de computación. Hay que ponderar lo que se desea, utilizar valores bajos en fase de pruebas y aumentarlos al final. El valor dado por defecto, 512 es bajo y habrá que aumentarlo probablemente hasta 2048 para obtener resultados adecuados. Pero sólo la práctica puede ayudar a decidir el valor adecuado. Al igual que ocurre con las texturas es recomendable utilizar tamaños que sean potencias de 2 (512, 1024, 2048...) pues las tarjetas gráficas están optimizadas para procesar estos tamaños.

b) el **Rango de muestreo** (*Sample Range*) que actúa como un *antialiasing* específico para sombras y afecta principalmente a la mayor o menor precisión de los bordes de la sombra. El valor predeterminado para el motor de render de 3ds Max es 4.0. Con valores superiores a 10.0 aparecerán claramente difuminados en la mayoría de los casos y con valores inferiores a 2.0 muy nítidos. Los parámetros de *mr shadow map* son similares aunque los rangos usan valores distintos, pero su significado básico es el mismo. El va-

lor predeterminado de *Sample Range* para *mr* es 0.01. Con valores superiores los contornos de las sombras se hacen más suaves (no hay límite superior). El valor asociado para *mr, Samples*, aumenta o disminuye el número de muestras que se toman de las sombras, lo que influye en la calidad del resultado.

c) El **bias** (desajuste o alteración). Este parámetro aumenta o disminuye la precisión de los cálculos, lo que afecta, principalmente, al ajuste entre la sombra y el objeto que la proyecta. Valores altos de *bias* (superiores a 1.0) aceleran el cálculo pero harán que la sombra se separe notoriamente de su objeto. Valores bajos (del orden de 0.1 o 0.01) acercan la sombra a su origen a costa de mayor tiempo de cálculo. Un valor muy bajo o igual a 0.0 podría hacer que se confundiera la región del espacio que está iluminada con la que no lo está, lo que podría dar lugar también a errores. Dado el aumento de velocidad de los ordenadores todas estas precauciones se están quedando obsoletas y se puede utilizar un valor muy bajo, tal como 0.01 sin que se aprecien disminuciones notables en los tiempos de cálculo.

Hay otros dos parámetros secundarios: *Alterar mapa absoluto* (*Absolute Map Bias*, desactivado por defecto) hace que el cálculo se haga a partir de los valores generales de la escena, normalizados a 1 (la precisión aumenta al acercarse a 0.01 a costa de mayor tiempo de cálculo y de que, si está muy próxi-

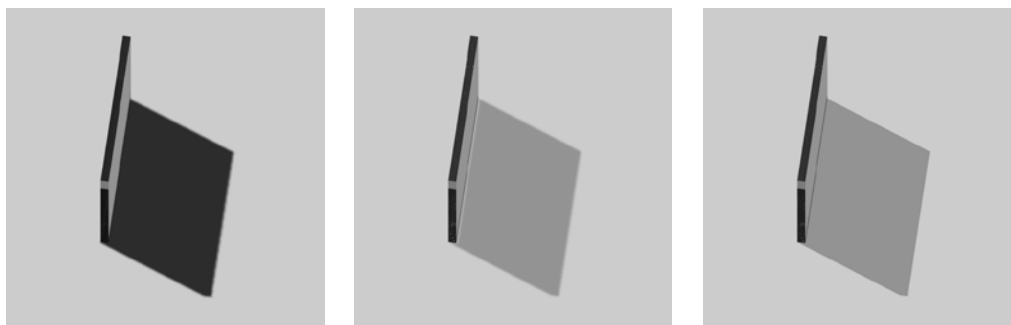


Figura 5.14 Transparencia: a) Sombras por mapa sin activar transparencia (4096, 0.2, 8: size, sample range, samples), b) Misma configuración con transparencia activada, c) Sombras por ray trace: la transparencia se reconoce automáticamente



mo a cero se produzcan errores) o bien, si se activa, a partir de las unidades genéricas utilizadas. Es preferible dejarlo como está. *Sombras de 2 lados* (*2 Sided Shadows*, desactivado por defecto) obliga a que se tengan en cuenta las caras ocultas. Cuando está desactivado, puede ocurrir que luces externas iluminen el interior de objetos o que caras situadas en el interior de objetos no arrojen sombras. Pero no suele ser muy recomendable trabajar con objetos de una sola cara por lo que carece de importancia en la práctica corriente. Si se utiliza el *mr shadow map* este parámetro está desactivado por defecto.

Las figuras adjuntas muestran los resultados de variar estos cuatro parámetros.

e) Hay un grupo adicional, **Sombras Transparentes** (*Transparent shadows*) para procesar objetos transparentes que en principio no son reconocidos por sombras por mapa (esta opción no existe en las sombras por mapa del motor predeterminado de 3ds Max ni en otros programas). Si se activa, quedan disponibles tres parámetros secundarios: *Color*: si se activa el color de la superficie afecta al color de la sombra (a costa de mayor tiempo de cálculo y ocupación de memoria), *Fusionar distancia* (*Merge Distance*): la distancia mínima que tiene que haber entre dos superficies para ser consideradas distintas, el valor predeterminado es 0.0. *Muestras por píxel* (*Samples/Pixel*): el número de muestras utilizadas para generar un píxel del mapa de sombras. El valor predeterminado es 5.

Sin embargo, si hay transparencias, es preferible utilizar sombras por *ray trace* que reconocen automáticamente los canales alfa de objetos transparentes, dan resultados más exactos e incluso pueden llegar a tardar menos si el tamaño del mapa es alto, como en la figura adjunta en que, la última, generada por *ray trace*, ha tardado poco más que el tiempo de un cálculo por mapa de sombra con un tamaño de 1024 a 4096 (el mismo tiempo entre estos extremos debido a la optimización de las tarjetas gráficas para este tipo de cálculos por mapa).

## Sombras por ray trace

La ventaja principal de este tipo es la precisión. Y sus principales desventajas, que genera sombras duras y que es algo más lento. El primer inconveniente, que genera sombras duras, desaparece si se utilizan luces de área. Si bien es cierto que el uso de luces de área agrava el segundo inconveniente, la lentitud. Pero este inconveniente se reduce cada año con los aumentos de prestaciones de los ordenadores aunque conviene recordarlo en escenas complejas y con muchas sombras.

Las sombras por *ray trace*, tanto del motor predeterminado de Max como de Mental Ray utilizan tres parámetros:

a) El **bias**, cuyo significado básico es el mismo que en el caso anterior, con un valor predeterminado de 0.01.

b) La **Máxima profundidad de árbol cuadrático** (*Map Quadtree Depth*, 7) que ajusta la profundidad del *Quadtree* (una estructura de datos utilizada en el cómputo de trazado de rayos que se subdivide sistemáticamente a medida que encuentra diferentes objetos en su rastreo) utilizado por el trazador: cuanto mayor sea este valor más exacto es el cálculo a costa de mayor tiempo de computación. Sólo tiene sentido aumentar su valor en esce-

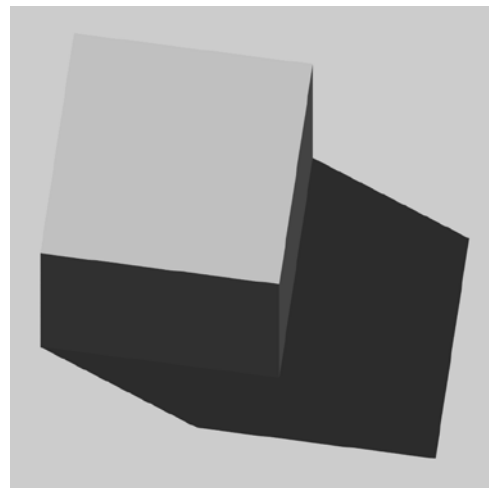


Figura 5.15 Sombras por Ray Trace



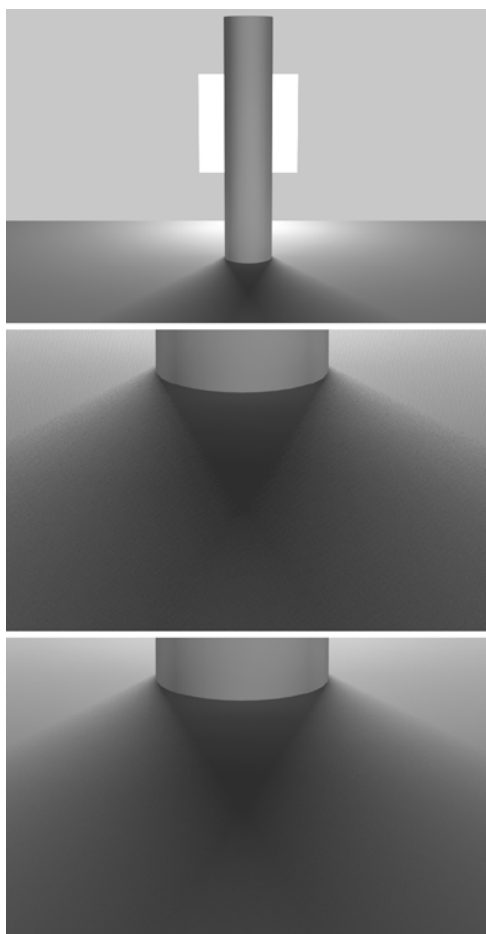


Figura 5.16 Sombras de área (a) y detalles: b) abajo a la izquierda uv samples: 3x3; c) abajo a la derecha, uv samples 8x8

nas muy complejas y si se detectan imperfecciones. Las luces omni utilizan 6 *quadtrees*, uno por cada dirección principal, una razón más por la que es muy recomendable no utilizarlas como luces principales.

c) El parámetro secundario, la casilla *Sombras de 2 lados* (2 Sided Shadows, desactivado por defecto) que tiene el mismo sentido que en el caso anterior.

## Sombras de área

Es posible simular sombras de área con luces puntuales, activando esta opción que está disponible en muchos programas de *render* y simula una distribución extensa aunque la luz sea propiamente puntual. Es uno de los tipos disponibles que he mencionado al comienzo de esta sección. Sin embargo los resultados son algo mejores con luces extensas y los ajustes son más sencillos.

El principal inconveniente de utilizar sombras de área es, como ya he dicho, que los tiempos de cálculo aumentan. Con luces extensas planas puede ser del orden de 1.6x. Y con luces de volumen del orden de 1.7x. Pero vuelvo a recordar que este inconveniente es cada vez menor con la creciente potencia de las tarjetas gráficas actuales.

Como ya hemos visto más arriba, si se utilizan **luces simples planas** en 3ds Max (*mr Area Spot*), aparecerá un nuevo grupo de parámetros, “parámetros de luz de área” (*area light parameters*) que permite elegir entre dos tipos, **disco** (con un parámetro para ajustar el radio) y **rectángulo** (con dos parámetros para

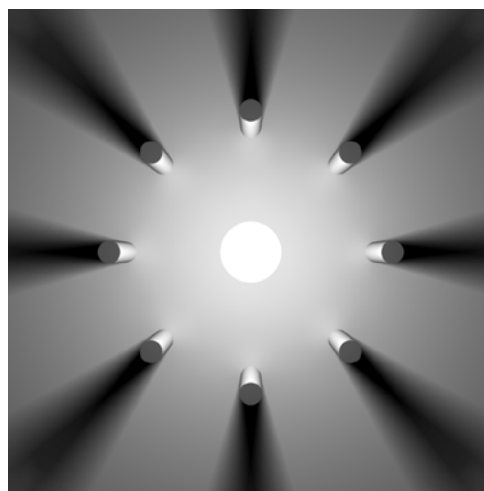


Figura 5.17 Sombras de área volumétricas producidas por una luz extensa, una esfera que ilumina por igual a los cilindros que la rodea



ajustar la anchura y altura). Y si se utilizan **luces simples de volumen** (mr Area Omni), aparecerá igualmente un nuevo grupo de parámetros, “parámetros de luz de área” (*area light parameters*) que permite elegir entre dos tipos, **esfera** (con un parámetro para ajustar el radio) y **cilindro** (con dos parámetros para ajustar el radio y la altura).

Como también he dicho, en sistemas de cálculo simple esto no tiene efecto apreciable en la distribución de la luz y sólo afectará a las sombras, que mostrarán efectos de umbra y penumbra. La figura adjunta muestra un cilindro iluminado por una luz de área. Se han utilizado sombras por *ray trace* pues los cálculos son más exactos y la zonas de umbra y penumbra se crean de modo natural.

Los parámetros específicos de luces de área son similares a los de las luces puntuales y no es necesario que los repitamos. El parámetro principal, para luces planas es *UV Samples* (predeterminado a 5×5) que afecta a la calidad de las sombras y, como siempre, convendrá reducirlo para pruebas y aumentarlo para resultados finales.

Las sombras producidas por luces con volumen, producen el mismo efecto que las planas pero son más adecuadas para el caso de luces que iluminen en todas direcciones. Para que las luces planas produzcan el efecto adecuado, el plano debe estar orientado perpendicularmente a la dirección de la luz.

Si se utilizan luces **fotométricas**, como también hemos visto nos encontramos directamente con tipos similares, planos y de volumen (*point*, *line*, *rectangle*, *disc*, *sphere*, *cylinder*) con los parámetros adecuados para ajustar las dimensiones correspondientes.

En este caso, el tipo de cálculo de sombras más recomendable es por *ray trace*. Los parámetros de ajuste de *ray trace* son básicamente los mismos.

Al escoger un tipo que no sea *point*, nos encontraremos con una nueva sección: *Shape/Area Shadows* que, además de utilizarse para escoger la forma (*point*, *rectangle*, *disc*, etc) incluyen un grupo con un parámetro im-

portante, *Shadow Samples* (32 por defecto) que es el equivalente de los *UV Samples* de las luces de área estándar. Con este parámetro se controla igualmente la calidad de las sombras aunque sus valores son distintos, son potencias de 2: 2, 4, 8, 16, 32, 64, 128, 256, 512 y 1024).

Según los casos habrá que aumentar este valor. Si se utilizan luces secundarias también convendrá jerarquizar este parámetro: las luces secundarias pueden quedarse en 16, las corrientes en 32 (el valor por defecto) y las luces principales que arrojen sombras muy visibles en 64. El control *Soft Shadow Precision* que se encuentra en la *Renderer Frame Window* aumenta los valores de *samples* de todas las luces pero mantiene esta jerarquía.

## Sombras de volumen

Las sombras de volumen (no confundir con las de *mr omni* que simulan luces extensas volumétricas) se basan en un algoritmo distinto que ya se ha explicado en el Cap 2.

Como ya hemos visto en aquél capítulo es una técnica alternativa que fue propuesta originalmente por Frank Crow en 1977. En 1991, Tim Heidmann presentó una variante que permitía implementar este tipo de cálculo por *hardware* por medio del *stencil buffer* lo que aceleró considerablemente el cálculo y contribuyó decisivamente a su popularización en ciertas aplicaciones de juegos de vídeo.

Sin embargo, pese a algunas ventajas notables, tales como el autosombreado automático o la precisión adaptable a la resolución entre otras, sus desventajas son también notables. Las principales son que requieren un alto coste de computación y GPUs especializadas, una reestructuración adecuada de la geometría de la escena y, sobre todo, que se basan en luces puntuales o direccionales por lo que no admiten luces de área. Por todas estas razones, por ahora, que yo sepa, no se utilizan en las aplicaciones corrientes a las que va dirigido este libro, por lo que las pasaré por alto.



## 5.4 Sistemas de luz diurna

### Procedimientos generales. Creación y especificación de datos climatológicos

Un “sistema de luz diurna” (*daylight system*) es una combinación de dos tipos de luz: luz solar (*sunlight*) y luz celeste (*skylight*). Estos dos tipos se agrupan de diferentes modos, según las aplicaciones. En 3ds Max están agrupadas como una *assembly*, un tipo específico de grupo para luces que permite, en principio, acceder a la intensidad sin abrir el grupo. En otros programas se mantienen independientes y los parámetros se especifican desde diferentes grupos de paneles.

En muchos casos, cada una de estas luces puede crearse por separado, tal como veremos en los siguientes apartados. Pero si se crean a partir de un sistema de luz diurna esto tiene la ventaja de que se pueden relacionar estas luces con bases de datos estadísticas que ajustan sus intensidades y otras características, a estos datos.

En este apartado describiré por tanto el procedimiento general, que abarca a ambos tipos de luces, y en los siguientes analizaré las características de cada uno de los dos tipos involucrados.

Para crear un sistema de luz diurna en 3ds Max el procedimiento es:

1. Ir al panel *Create/Systems* y escoger *Daylight* (o ir al menú *Create/Lights/Daylight*).
2. Si aparece un cuadro de diálogo con el mensaje de que se recomienda utilizar un *mr Photographic Exposure Control* con un valor de EV = 15, aceptar.
3. En un visor adecuado pinchar y arrastrar: así se define la rosa de los vientos que indica la orientación.
4. Si aparece otro mensaje recomendando utilizar un *mr Physical Sky* como mapa de entorno, responder también que sí.
5. Soltar el botón del ratón y seguir moviéndolo: así se define la posición de la luz.

Pueden definirse la orientación y las propiedades en el momento de la creación o posteriormente, desde el panel *Modify*. Para abreviar, explicaré el procedimiento de modificación, a posteriori, que incluye más variantes.

Para modificar un sistema de luz diurna seleccionarlo y:

1. En *Daylight Parameters*, en el grupo *Position*, escoger el modo de asignación de la posición del sistema. Hay tres variantes: *Manual*, *Date-Time-Location* y *Weather Data File*. El primero se explica por sí solo: el usuario puede asignar libremente la posición moviendo el sistema como un objeto más de la escena. El segundo sitúa el sistema en función de las coordenadas geográficas, el día y la hora mediante un cuadro de diálogo específico. El tercero es similar pero las asigna a partir de un fichero de datos que se utiliza para asignar propiedades adicionales. Escoger la segunda opción. La tercera la explicaré más abajo.
2. Después de escoger la opción *Date-Time-Location*, presionar el botón *Setup*, que lleva al panel *Motion*, o entrar directamente en este panel (en el caso de una luz solar no hay botón de *setup* y hay que entrar directamente en este panel: estos parámetros se utilizan en animación, razón por la que están en el panel *Motion*). Cambiar la fecha y la hora desde el grupo *Time*. Cambiar la posición geográfica desde el grupo *Location*.
3. Desde este mismo panel se puede modificar la orientación (la dirección del norte). Para ello, modificar los valores situados junto a *North Direction* y especificar un valor distinto de 0 para cambiar el ángulo de la rosa de los vientos. En este mismo grupo se puede también cambiar la distancia de la luz a la rosa de los vientos, modificando el valor de escala orbital: esto sólo afecta a la organización de los objetos en la escena, no a la intensidad de la luz.
4. Si lo que se quiere modificar es el tamaño o la visibilidad de la rosa de los vientos



(compass), seleccionarla e ir al panel *Modify*. Aunque lo mejor será ocultarla y olvidarse de ella.

5. Si lo que se quiere modificar son parámetros generales de la luz (intensidad, color, atenuación, tipo de sombras, etc.) ir al panel *Modify* y cambiar estos parámetros que variarán según el tipo de luz solar que hayamos escogido y que se describen el siguiente apartado. En principio no es muy buena idea modificar los valores de intensidad pues estos valores se ajustan automáticamente para reflejar los propios de las condiciones geográficas y temporales escogidas según cuadros estadísticos internos.

La otra alternativa que he mencionado al comienzo es *Weather Data File*. Esta opción debe escogerse únicamente si se van a utilizar los modelos de cielo disponibles para *mr sky* que veremos más adelante. En este caso, al presionar el botón *Setup* se entra directamente en un cuadro de diálogo específico en donde lo primero que se pide es cargar un fichero de datos. El procedimiento que habría que seguir en este caso sería el siguiente.

1. Previamente, desde Internet, ir a la página oficial, del Departamento de Energía

de Estados Unidos, *EnergyPlus Energy Simulation Software* ([http://www.eere.energy.gov/buildings/energyplus/cfm/weather\\_data.cfm](http://www.eere.energy.gov/buildings/energyplus/cfm/weather_data.cfm)) y, de aquí, a Europa, España, Barcelona (o cualquier otro lugar del mundo que interese). Descargar el archivo de datos en formato EPW (*Energy Plus Weather format*) correspondiente a la región escogida.

2. Guardar este archivo en un lugar adecuado, fácilmente accesible desde el programa que vayamos a utilizar.
3. Seleccionar el sistema de luz diurna y en *Daylight Parameters*, escoger como el modo de asignación de la posición la opción *Weather Data File*.
4. Presionar el botón *Setup* para entrar en el cuadro de diálogo *Configure Weather Data*. Presionar el botón *Load Weather Data* y escoger el fichero que hemos descargado. Con esto quedarán disponibles los controles para ajustar el mes, día y hora presionando el botón *Change Time Period*. El año no se puede cambiar: es el año en que se obtuvieron los datos, 1989.
5. Al escoger esta opción, los valores de iluminación para la región seleccionada quedarán disponibles para los modelos de cielo que veremos más adelante.

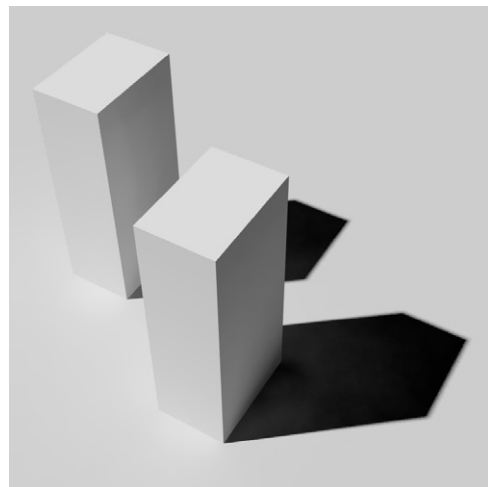
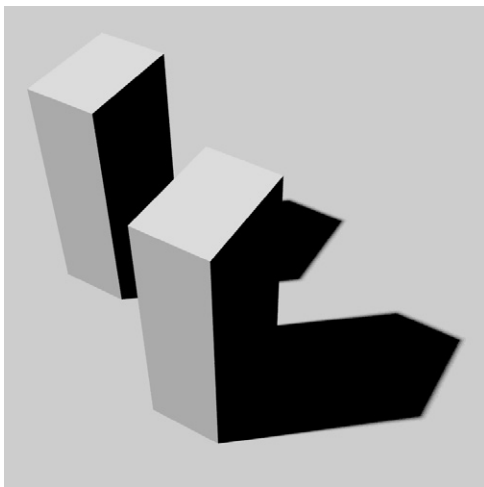
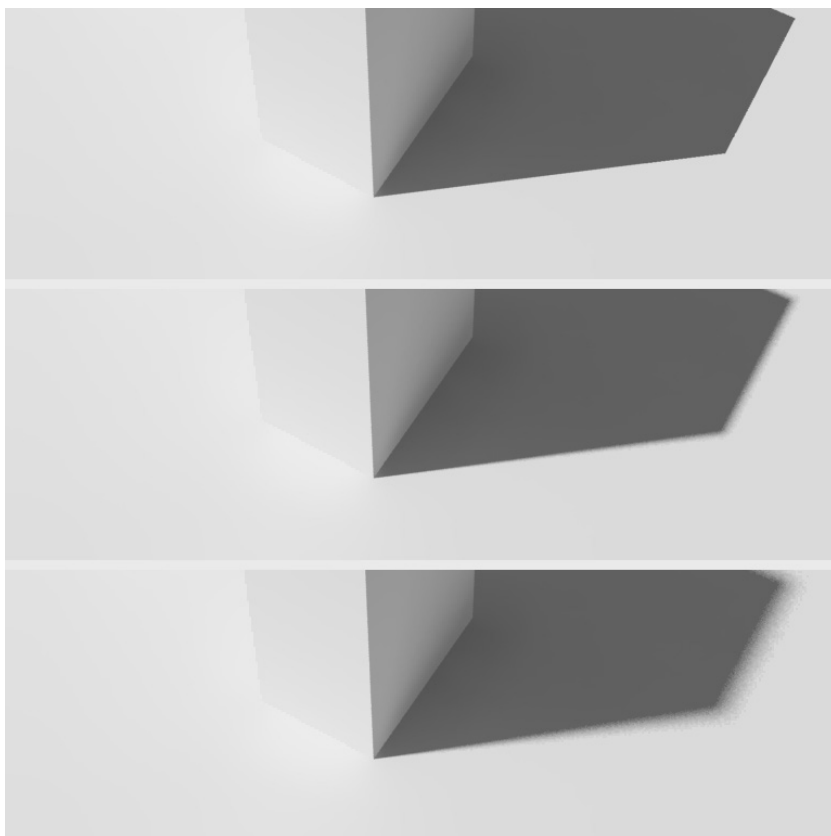
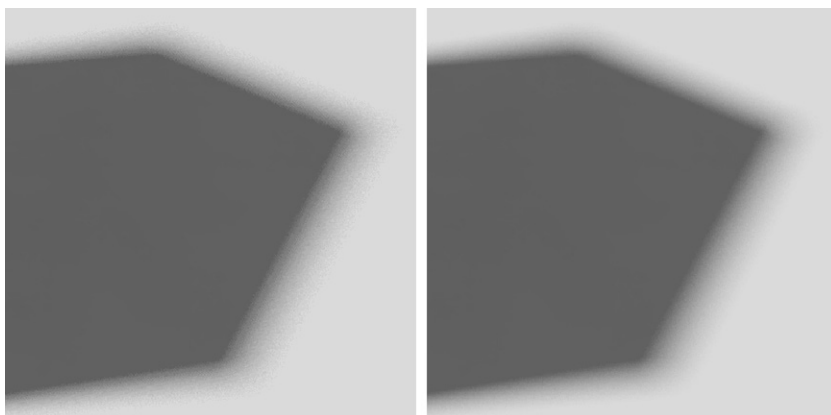


Figura 5.18 a) Luz solar simple, sin iluminación avanzada (EV 14), b) Luz solar simple con iluminación avanzada (FG medium, EV 14)



*Figura 5.19 Sombras de mr sun: a) Softness 0.0, b) Softness 1.0, c) Softness 2.2 samples 16*



*Figura 5.20 Sombras de mr Sun. Detalle: a) samples 8, b) samples 32*



## Luz solar. Tipos y propiedades

La luz del sol se puede simular por medio de una luz directa simple (estándar), como ya hemos visto en los apartados anteriores correspondientes a luces simples. Pero los parámetros son muy limitados y las unidades arbitrarias. Para simular una luz con unidades reales y propiedades avanzadas que puedan procesarse adecuadamente por sistemas de iluminación avanzada es preferible utilizar una luz solar específica que, en 3ds Max, se encuentra en el panel *Create/Systems*. Se puede escoger *Sunlight* o *Daylight*. El procedimiento es el mismo en ambos casos y ya se ha descrito en el apartado anterior. La única diferencia es que en el primer caso se crea una única luz y en el segundo caso se crean dos (una *sunlight* y una *skylight*) Pero como cada una de estas se puede desactivar, en general será más práctico, para crear una luz solar independiente, crear una luz diurna y desactivar el componente celeste.

Los tipos de luz solar que se encuentran en 3ds Max son los siguientes:

**Standard.** Los parámetros disponibles son los de una luz simple. La intensidad vendrá dada por un multiplicador que traduce los valores dados en unidades reales (luxes) a un valor equivalente que será del orden de 3.0 para 60.000 luxes, 4.0 para 80.000 o 5.0 para 100.000.

**IES Sun.** Los parámetros disponibles son similares a los de una luz simple pero los valores de intensidad, en la sección *Sun Parameters*, se dan en luxes y corresponden a los valores medios estadísticos para la zona geográfica, el día y la hora especificados. Los valores característicos suelen ser del orden de 80.000 o 90.000 luxes. El color de la luz también se ajusta a la temperatura de color correspondiente a esas condiciones.

**mr Sun.** Los parámetros disponibles se modifican completamente y sólo aparecen dos secciones: parámetros básicos y fotones. La

primera sección, *mr Sun Basic Parameters*, incluye controles habituales para activar o desactivar la luz y un multiplicador cuyo valor inicial es 1.0. Este valor está relacionado con los valores medios estadísticos por lo que, si no se toca, el resultado será el mismo que en el tipo anterior, el correspondiente a las estadísticas de la zona. Y si se aumenta o disminuye se alterará este valor. Las sombras se controlan mediante parámetros especiales que comentaré en la sección sobre sombras pero que, en pocas palabras, controla la amplitud de la zona de penumbra a medida que las sombras se alejan del objeto que las causa, como ocurre en los casos reales. La segunda sección tiene que ver con el cálculo de Iluminación Avanzada por medio de fotones y también se comentará en el capítulo siguiente.

Las diferencias entre utilizar uno u otro tipo de luz se manifiestan de un modo notorio en las sombras.

Las sombras de una luz simple (*standard*) carecen de matices. Las sombras de *mr sun* son similares a las que se consiguen con *mr area* pero con la ventaja de que están incorporados al sistema de luz diurna. El ajuste de sombras se realiza desde la sección *mr Sun Basic Parameters*, en el grupo *Shadows*. Este grupo incluye tres parámetros principales:

a) *On/Off*. Activa o desactiva las sombras,  
 b) *Softness* (1.0). Suaviza los bordes de las sombras en función de la mayor o menor distancia con respecto al objeto que las produce, como ocurre con las sombras reales. Con un valor de 0.0 no hay suavizado y el resultado es similar al obtenido con *ray traced shadows*. Con el valor predeterminado hay un ligero suavizado. Con valores superiores, por ejemplo 4.0, el suavizado es notorio y aparecerán defectos, ruido. Para corregirlos, aumentar el valor del siguiente parámetro.

c) *Softness Samples* (8.0). Este parámetro aumenta el número de muestras y reduce el efecto de ruido cuando el suavizado aumenta. Pero debe tenerse en cuenta que esto también incrementa el tiempo de cálculo.





Las figuras adjuntas ilustran estas diferencias, tanto por lo que respecta a una luz solar estándar como a las diferentes configuraciones de una luz solar de mental ray.

Por otro lado, el color de la sombra se ve afectado por las características del mapa procedural de cielo, *mr Physical Sky* que también se ha comentado más arriba y que puede resultar algo azulado. Si se mantiene activada la casilla *Inherit from mr Sky* (heredar de *mr sky*) los ajustes de color y contraste derivarán de los asignados a la atmósfera general (*sky*). Si se desactiva, se pueden ajustar de modo independiente aunque esto no es muy recomendable pues complica la gestión del sistema de luz diurna. En cualquier caso los dos parámetros disponibles, *Red/Blue Tint* y *Saturation* permiten ajustar el color de la sombra. Lo más sencillo suele ser rebajar el valor de este último. Con un valor tal como 0.2 resulta un gris prácticamente neutro. Si se modifican los valores del primero se altera la temperatura global de color de la escena. Valores negativos harán que la tonalidad general sea más fría, azulada. Valores positivos harán que sea más cálida, rojiza. Si se introducen modificaciones en estos valores convendrá utilizar variantes mínimas para no alterar demasiado la tonalidad de la escena. Obviamente, esto

depende de los casos y de las intenciones del usuario. Pero en caso de duda lo mejor será no tocarlos.

### Luz celeste. Tipos y propiedades

La luz dispersada por la atmósfera se puede simular por medio de un tipo especial de luz, una luz celeste (*skylight*). Hay dos tipos básicos: una luz celeste simple (*standard*) y una luz celeste que utiliza modelos relacionados con unidades reales y valores estadísticos, si bien en este caso las posibilidades son más complejas pues, como hemos visto en el capítulo 1, la representación de la distribución de luminancias de la atmósfera es un tema bastante complicado para el que hay varias alternativas y nos encontraremos, también como en el caso anterior, con tres subtipos principales.

El procedimiento de creación de una **luz celeste estándar** es muy sencillo pues todo lo que hay que hacer es pinchar en un punto de un visor en planta o en perspectiva para que se cree sobre el plano horizontal de la escena. Aparecerá un icono en forma de semiesfera apuntando hacia abajo. La posición concreta en que se cree es irrelevante pues una luz cenital no es sino una especie de luz

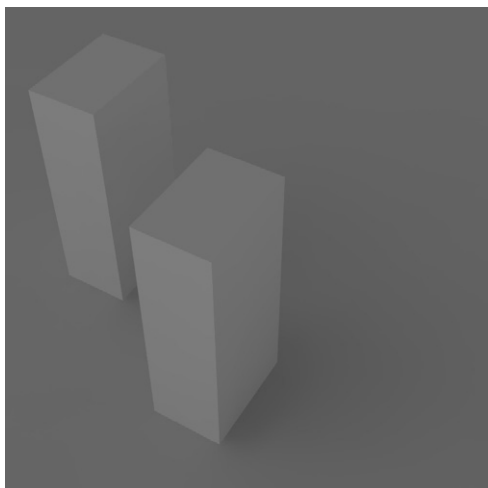


Figura 5.21 Luz celeste con iluminación avanzada (FG medium, EV 14)

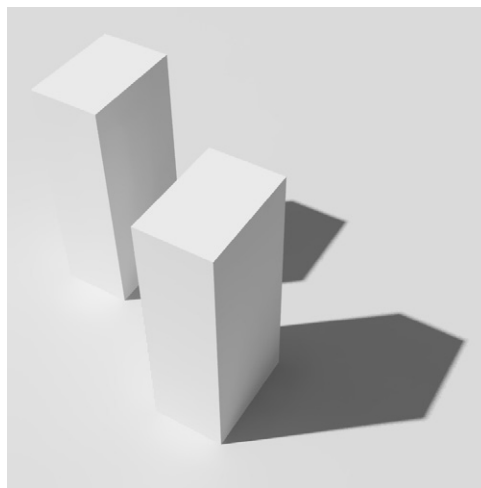


Figura 5.22 Luz solar y luz celeste combinadas (FG medium, EV 14)



ambiente que ilumina por igual en todas las direcciones correspondientes a un hemisferio que abarcaría toda la escena y con luminosidad homogénea.

La luz celeste está pensada para ser utilizada con sistemas de iluminación avanzada relativamente sencillos. Por si sola no produce otra cosa que una iluminación uniforme semejante a la de la vieja luz de tipo *ambient*.

Una vez creada se pueden modificar sus parámetros: *On/Off* para activarla o desactivarla; *Multiplicador* de intensidad (1.0 predefinido, como todas las luces simples); el grupo *Color* permite asignarle un color específico, hacer que sea el mismo definido para el fondo (en el menú *Environment*) o enlazarlo a un mapa, como veremos más adelante. Además hay un grupo, *Render*, que permitiría activar la opción de calcular sombras cenitales, independientemente, para este tipo de luz pero que no conviene utilizar pues el tiempo de *render* se dispara inutilmente (se mantiene por compatibilidad con métodos antiguos de cálculo de iluminación semi-avanzada). Si se quiere trabajar sólo con esta luz hay que utilizar sistemas de iluminación avanzada específicos que veremos en el siguiente capítulo.

Si se crea una **luz celeste integrada en un sistema de luz diurna** también se crea automáticamente una luz cenital pero con propiedades avanzadas. En este caso el icono aparecerá ligado a la luz solar aunque el efecto sobre la escena es el mismo.

Los tipos disponibles en este caso son los siguientes:

**Skylight.** Los parámetros disponibles son los mismos que los de una luz celeste simple (*standard skylight*) que acabo de describir. Las propiedades son independientes de los valores dados por las condiciones (coordenadas geográficas, fecha y hora). Al igual que en el caso anterior, rara vez se utiliza este tipo aislado. En versiones anteriores de 3ds Max se podía utilizar con un sistema simple de iluminación avanzada, *Light Tracer*, que sigue estando disponible pero que no tiene ninguna ventaja con respecto a los que veremos más adelante.

**IES Sky.** Los parámetros disponibles son similares a los anteriores pero hay un grupo adicional, *Coverage*, que permite elegir entre tres modos de dispersión de la luz en la atmósfera: cielo despejado (*clear*), parcialmente cubierto (*partly cloud*) y cubierto (*cloudy*). Pero las diferencias, como veremos, son poco apreciables.

**mr Sky.** Si se elige esta opción aparecen nuevas secciones. Las principales son las tres siguientes:

a) *mr sky parameters*, que incluye parámetros similares a los anteriores (*on/of*, multiplicador) más un parámetro adicional para el color del terreno y otro para escoger un modelo de cielo (*sky model*) que da acceso a tres opciones: *Haze driven*, *CIE* y *Perez All Weather* que se describirán en el siguiente apartado;

b) Parámetros propios del modelo escogido (*Haze*, *CIE*, *Perez*) y que también se discuten en el siguiente apartado;

c) *mr Sky Advanced Parameters*. Esta sección incluye un parámetro para controlar la altura del horizonte, *Height* (0.0) y la nitidez de la transición entre el cielo y el terreno, *Blur* (0.1). También puede especificarse el color del cielo de noche con *Night Color* (negro por defecto). El grupo *Non-Physical Tuning* se utiliza para ajustar el color del cielo por medio de los parámetros que ya hemos visto en el caso de *mr sun*. Pero si, en *mr Sun*, se marca la opción "Inherit from mr sky", la temperatura de color (con *Red/Blue Tint*) y la saturación (con *Saturation* (1.0) se controlan desde aquí. Como ya he dicho más arriba, puede ser conveniente rebajar un poco la saturación pues a menudo la contribución de *mr Physical Sky* resulta excesivamente azul y, por otro lado, es recomendable no tocar la temperatura de color a no ser que se tenga muy claro que interesa hacerlo por alguna razón.

Por último, si se marca la casilla *Aerial Perspective* la visibilidad disminuye en función de la distancia.

## Luz celeste. Modelos de cielos

Si se utiliza una luz celeste de mental ray (*mr skylight*) hay tres modelos de cielo disponibles:

El modelo **Haze Driven** se basa en el número de partículas que puede haber en el aire (vapor de agua, polvo, partículas diversas). Los valores van de 0.0 (sin partículas) a 15.0 (cielo cubierto por niebla o por una tormenta de arena). Afecta a la intensidad y el color de todos los objetos, las sombras, el aspecto del fondo y la perspectiva aérea. Estos aspectos se verán más adelante, en el apartado correspondiente. Es una opción práctica que da buenos resultados pero no es adecuada si se quiere obtener una representación física precisa o utilizar la herramienta de *lighting analysis* que también veremos más adelante.

El modelo **CIE** se ha descrito en el capítulo 1 y corresponde a las normativas de la CIE (*Comission International de L'Eclairage*) para dos casos extremos normalizados, un cielo cubierto y un cielo despejado. Utiliza los mismos parámetros que el anterior, con los mismos valores por defecto, y con parámetros adicionales para escoger entre un cielo cubierto (*overcast*) o despejado (*clear*). Por defecto está activado *overcast*. El parámetro *Diffuse Horizon* (10.000 lx por defecto) configura la cantidad de luz que llegaría desde el horizonte, sin tener en cuenta la luz directa del sol. Es un valor estadístico que depende de la latitud: sería del orden de 15.000 a 18.000 lx en el ecuador e iría descendiendo hacia el norte. El valor predeterminado correspondería a una latitud de unos 50° (sur de Inglaterra). El parámetro *Direct Normal Illumination*

configura la cantidad de luz que vendría de la dirección del sol. Dado que es difícil tomar este tipo de decisiones que acaban por ser arbitrarias, es mejor utilizar un *weather data file*.

De todos modos debe tenerse en cuenta que los datos estadísticos son muy variables según las fuentes consultadas. Las imágenes que siguen están obtenidas a partir de configuraciones que han tomado valores medios generales para las fechas y horas indicadas. Para no tener que ir abriendo y cerrando la configuración de *Weather Data* se puede configurar para que muestre los valores en un rango de tiempo. Marcar la opción *Display Data as Animation* y dar una hora de inicio (*start*) y final (*end*) y el número de marcos (*frames*) que se mostrarán. Luego presionar el botón *Match Time*. Así podemos anotar una serie de valores de referencia.

Las recomendaciones dadas para el caso anterior también valen para este modelo, incluyendo la reducción del EV a valores comprendidos entre 13 y 14 para compensar cierta pérdida de luminosidad.

El modelo **Perez All Weather** se ha descrito también en el capítulo 1. Es un modelo basado en mediciones físicas estadísticas para casos intermedios entre cielos cubiertos y despejados. La versión que se incorpora a 3ds Max es una versión simplificada del modelo que he resumido en dicho capítulo y utiliza dos parámetros: *Diffuse Horizon* (10.000 luxes) que correspondería a la medición horizontal excluyendo el sol y *Direct Normal Illumination* (10.000 luxes) que correspondería a la medición dirigida hacia el sol. Da resultados físicamente correctos y es reconocido como un estándar industrial. Es adecuado para

hora	1 enero		1 abril		1 julio		1 octubre	
8:00	2.243	0	29.565	31.373	55.535	56.769	30.662	38.188
12:00	17.492	4.315	48.765	85.961	64.853	82.712	42.730	75.818
16:00	4.095	2.379	41.238	62.150	59.550	68.439	30.075	57.656

Figura 5.23 Valores correspondientes a las fechas y horas indicadas con la configuración Weather Data File y los datos dados por un archivo EPW correspondiente a Barcelona. La primera columna indica los valores del cielo que no reciben iluminación directa del sol (*Diffuse Horizon*) y la segunda los que sí la reciben (*Direct Normal*)

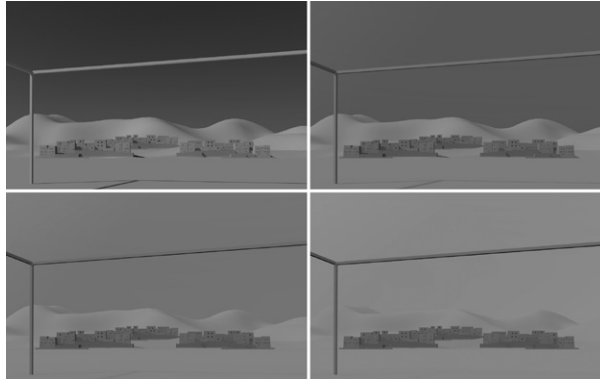


Figura 5.24 Skylight, modelo Haze Driven. De izquierda a derecha y de arriba abajo: a) Haze 0.0, b) 5.0, c) 10.0, d) 15.0. No se muestra el color que varía como sigue: azul intenso (a), azul grisáceo (b), verde amarillento (c), amarillo verdoso (d)

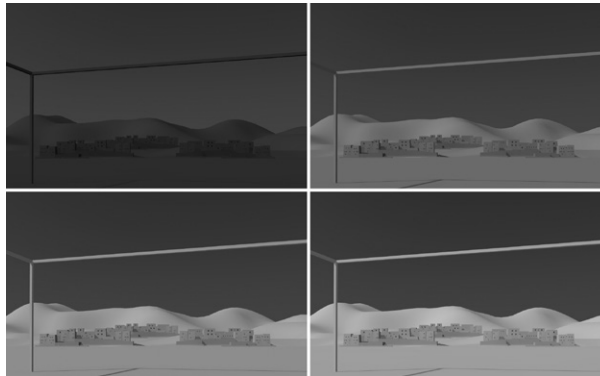


Figura 5.25 Skylight, modelo CIE, clear. Mismas condiciones que el ejemplo anterior. Valores para diffuse horizon/ normaldirect: a) 15.000/ 5.000 (invierno), b) 30.000/50.000 (otoño), c) 45.000/70.000 (primavera), d) 60.000/90.000 (verano). No se muestra el color que es azul en todos los casos

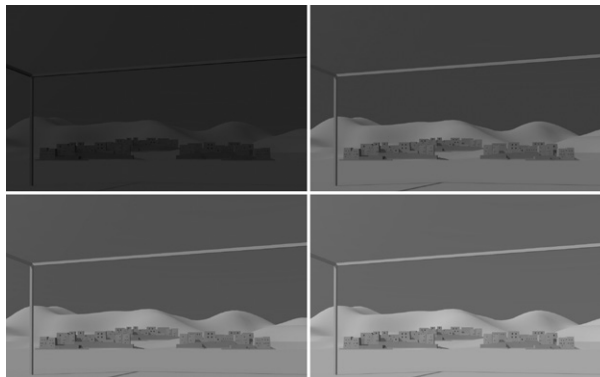


Figura 5.26 Skylight, modelo Perez All-Weather. Mismas condiciones que el ejemplo anterior y mismos valores (a) 15.000/ 5.000, b) 30.000/50.000, c) 45.000/70.000, d) 60.000/90.000). No se muestra el color que es azulado en todos los casos



escenas diurnas pero no tanto para horas tempranas o tardías (amanecer, crepúsculo, noche). Conviene utilizarlo para mediciones de exteriores o interiores con la herramienta *lighting analysis* pues esta herramienta está calibrada con este modelo. Puede ser que haya que reducir algo el EV del control de exposición.

Para probar estos modelos hacer lo siguiente:

- 1 Crear y configurar un sistema de luz diurna tal como se ha explicado más arriba. Escoger como tipo de luz solar *mr sun* y como tipo de luz celeste *mr sky*. En parámetros de *mr sky* escoger como *sky model* el que se indique en las figuras adjuntas (*CIE* o *Perez-All Weather*) y en parámetros avanzados especificar valores similares a los de las figuras adjuntas.
- 2 Configurar el cálculo de iluminación con los siguientes valores: Calidad 1/4-1 (min-max). *Final Gather* activado en modo "medium" con 2 rebotes (*bounces*). Ver el capítulo siguiente para una explicación más detallada de estos valores.
- 3 Ajustar el Control de Exposición a un EV 14 aproximadamente.

Para configurar el sistema de luz diurna de modo que muestre los datos estadísticos para un modelo de cielo intermedio durante un determinado periodo, y utilizar estos valores como referencias aproximadas, hacer lo siguiente:

- 1 Asegurarse de que contamos con un archivo de datos en formato EPW correspondiente a la región que vamos a analizar, tal como he explicado más arriba, en la introducción general a sistemas de luz diurna.
- 2 Seleccionar el sistema de luz diurna y en *Daylight Parameters*, escoger como el modo de asignación de la posición la opción *Weather Data File*. Presionar el botón *Setup* para entrar en el cuadro de diálogo *Configure Weather Data*. Presionar el botón *Load Weather Data* y escoger el fichero

que hemos descargado. Con esto quedarán disponibles los controles para ajustar el mes, día y hora.

- 3 En lugar de escoger la opción "Use Specific Date/Time", como antes, escoger la opción *Display Data as Animation*. Presionar el botón *Change Time Period* junto a *Start* y escoger la fecha y hora, 1 de enero, 09:00. Luego presionar el mismo botón junto a *End* y escoger la misma fecha pero distinta hora, 1 de enero, 17:00. En la casilla junto a "One Frame" escoger *Period* para que cree tantas claves de animación como horas del periodo considerado. A la derecha, encima del botón *Match Timeline* aparecerá el número de *Frames* correspondiente al periodo que hemos dado (9). Presionar el botón *Match Timeline* para que la línea del tiempo de la animación se ajuste a estos *frames*. Confirmar para salir.
- 4 La línea del tiempo de la animación habrá cambiado para mostrar tan sólo 9 frames. Al presionar el botón *Next Frame*, del grupo de controles de la animación, el sol saltará a la posición correspondiente a la siguiente hora. Con el sistema de luz diurna seleccionado, si se abre el panel *Motion*, se verá la hora correspondiente. Y si se abre el panel *Modify mr Sky CIE Parameters* (o *mr Sky Perez All Weather Parameters*, según cuál se haya seleccionado) se verán los valores correspondientes de los parámetros *Diffuse Horizon* y *Direct Normal Illuminance*. Anotar los valores correspondientes y, si luego se repite la misma operación para el 1 de marzo, el 1 de junio y el 1 de diciembre se obtendrán una serie de valores como los de la tabla adjunta.

Deben tenerse en cuenta los desfases entre la hora real y los horarios de verano. En nuestra zona horaria (España y Europa central), la hora corriente es GMT/UTC+1 (GMT son las siglas de *Greenwich Mean Time*, referidas al meridiano de Greenwich y UTC las siglas de *Universal Time-Coordinated*, referidas a relojes atómicos). Pero por acuerdo



de la UE (directiva 200/84) todos los países adelantan sus relojes 1 hora el último domingo de marzo y la retrasan el último domingo de octubre. Los valores anteriores deben tener en cuenta este dato pues, por ejemplo, las 16:00 de julio en Barcelona serían “en realidad” las 17:00, dado que en julio estamos en GMT/UTC+2.

Las opciones despejado (*clear*) o cubierto (*overcast*) se controlan mejor ajustando los valores correspondientes a la iluminación directa. Si este parámetro se hace 0.0, equivale a especificar que el cielo está completamente cubierto y toda la iluminación corre a cargo del parámetro difuso. Y si se le asigna un valor intermedio se controla el grado de reducción de la iluminación. Las figuras adjuntas ilustran estas variaciones para tres casos simples que corresponden a una misma hora pero con diferentes parámetros para el modelo de cielo.

Los resultados, desde el punto de vista de la iluminación, son similares si se utiliza el modelo *Perez All-Weather* pero la influencia del mapa de fondo afecta más al color, como puede apreciarse en la figura adjunta que se ha procesado con los mismos parámetros que la anterior.

Estas diferencias se pueden ajustar con el control de exposición (que haría más clara o más oscura la imagen) y los parámetros de *Saturation* y *Red/Blue tint* que se encuentran en la sección *mr Sky Advanced Parameters*.

## Perspectiva aérea

La simulación de efectos de niebla se lleva a cabo por procedimientos especiales que abarcan todo tipo de casos (humedad, contaminación, humo, etc) y que se describen más extensamente en el libro sobre Simulación de Materiales.

El efecto de “perspectiva aérea”, que ya se ha analizado en el Capítulo 1 puede simularse con más facilidad mediante parámetros que están incorporados a las propiedades de la *Sky Light* que se encuentran en el apartado de propiedades avanzadas y se especifican mediante un valor de distancia que es relativo a la visibilidad a la que se percibiría un 10% de la niebla (*haze*). Los valores adecuados dependen de los casos. En las figuras adjuntas se han mostrado los resultados que se obtendrían variando los valores del parámetro de *haze*. Con valores bajos, los edificios y las montañas del fondo son apenas perceptibles.

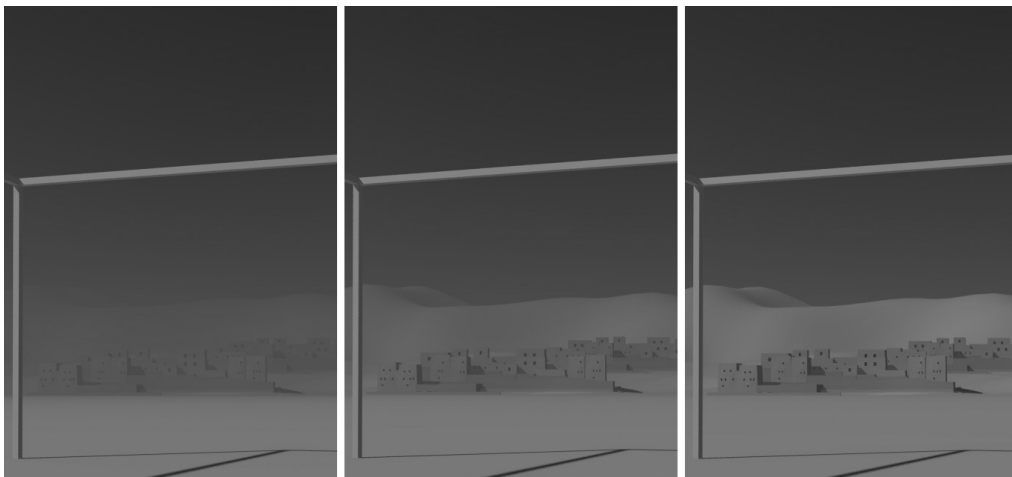


Figura 5.27 *Sky Light*. Perspectiva aérea. Diferentes valores de distancia de visibilidad como 10% del valor de haze (para haze 0.0): a) 30, b) 60, c) 120



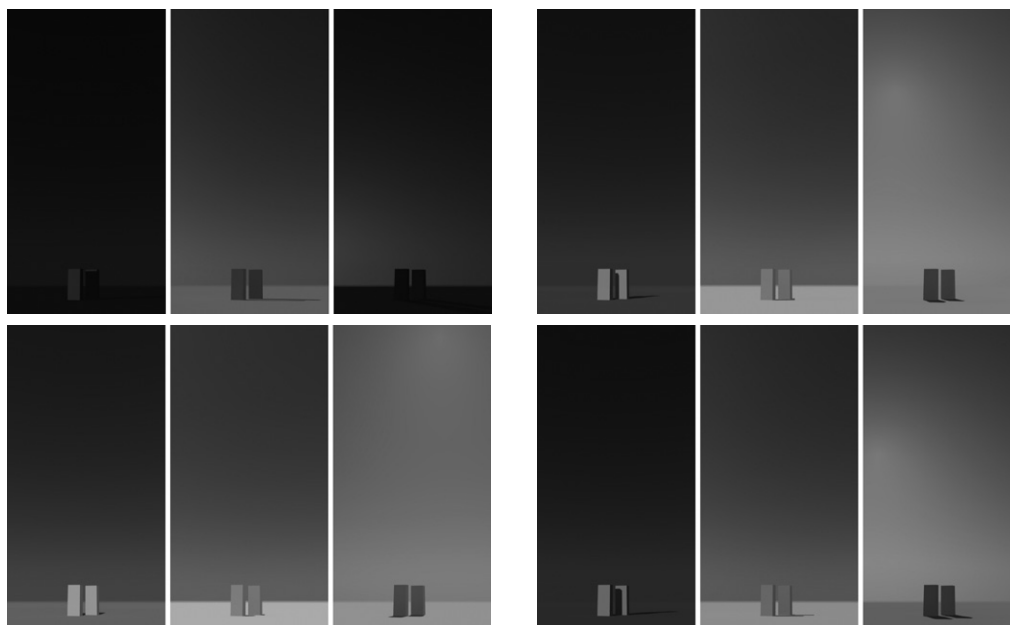


Figura 5.28 Mapa mrPhysical Sky. De izquierda a derecha y de arriba abajo (no se muestra el color que es más cálido al atardecer):

- a) 1 de enero, 8:00, 12:00, 16:00pm;
- b) 1 de abril, 8:00, 12:00, 16:00pm;
- c) 1 de julio, 8:00, 12:00, 16:00pm;
- d) 1 de octubre, 8:00, 12:00, 16:00pm

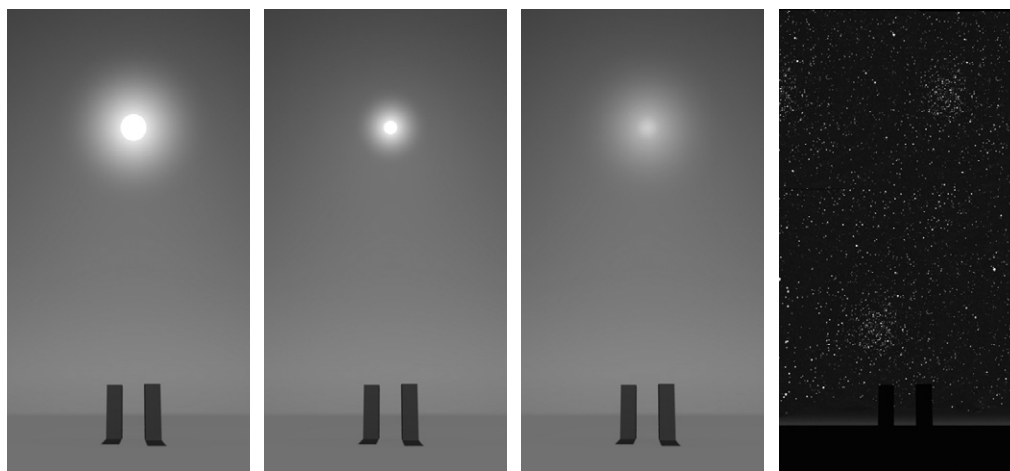


Figura 5.29 Diferentes aspectos para el sol, 1 de junio, 5:00pm (visto desde la cámara situada en el Este) Intensidad del disco/ Intensidad del brillo/Escala: a) 1.0/1.0/4.0, b) 1.0/1.0/2.0, c) 0.001/0.04/4.0; d) Fondo de estrellas



## Mapas de entorno

El uso de mapas de fondo se trata ampliamente en el libro sobre Simulación de Materiales por lo que, de nuevo, me remito a los capítulos correspondientes de ese libro. En este apartado me limitaré a algunas referencias básicas necesarias para la simulación de la iluminación.

Al crear un sistema de luz diurna se crea automáticamente, con *mental ray*, un mapa de tipo *mr Physical Sky* que se asigna como fondo a toda la escena. En otros programas se utilizan recursos similares.

El uso de este mapa tiene importantes ventajas. La principal es que es va ligado a un algoritmo que hace que el color del cielo cambie de color y de intensidad en función de la posición del sol, lo que nos ahorra muchas operaciones de ajuste. Las figuras adjuntas muestran algunas variaciones características. Si el lector reproduce estas configuraciones en un monitor podrá apreciar mejor los cambios de intensidad y de color.

Para utilizar este mapa, en la mayoría de los casos no hay que hacer nada, pues se carga automáticamente. Pero en algunos casos puede interesar hacer algún ajuste adicional. Para editarlo, arrastrarlo desde el cuadro de diálogo *Environment and Effects* a

un visor del Editor de Materiales. Si se hubiera cambiado por otro mapa y se quisiera volver a cargarlo o se hubiera descargado por alguna razón, todo lo que hay que hacer es abrir el Editor de Materiales, seleccionar un visor cualquiera, presionar el botón *Get Material* para entrar en el *Material/Map Browser* y buscarlo en la sección *Maps/Mental Ray*. Luego, desde el Editor de Materiales, hacer lo contrario que para editarlo: arrastrarlo sobre el botón *Environment Map* del cuadro de diálogo *Environment and Effects*. En cualquiera de los dos casos, al arrastrar el mapa se abrirá un pequeño panel que preguntará si lo queremos duplicar como “copia” o como “instancia”. Escoger instancia para que cualquier cambio se transmita automáticamente del editor de materiales a la proyección de entorno.

Al editarlo, aparecerán una serie de propiedades que se pueden dividir en tres grupos principales: a) las que afectan al aspecto del sol, en caso de que sea visible, b) la que ofrece la posibilidad de utilizar otro mapa, lo que es equivalente a utilizar un mapa de fondo del modo habitual, con la única ventaja de que, al estar situado en este cuadro de diálogo, se puede activar o desactivar con facilidad substituyéndolo por el *physical sky* sin tener que repetir la operación de asignación, c) una serie de propiedades que son similares a las

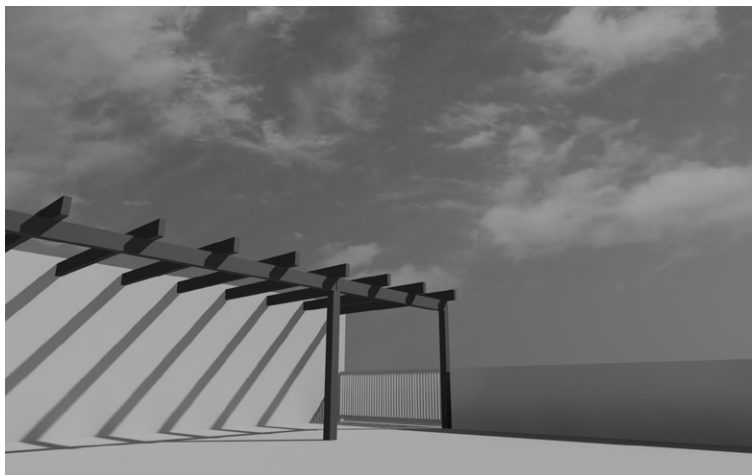


Figura 5.30 Combinación de un mapa de bits con *mr Physical Sky*



que ya hemos visto en *mr sky* pero con alguna adición interesante: además de especificar el color del terreno, la altura del horizonte, etc, se puede especificar un mapa para el terreno y también se puede especificar un color o un mapa para el cielo nocturno. Por defecto está activada la casilla “Inherit from mr sky” de tal modo que estas propiedades se heredan de lo especificado en los parámetros de *mr sky*. Pero si se desactiva esta casilla se pueden utilizar estas propiedades adicionales.

Una opción particularmente interesante es combinar un mapa de bits con un *mr Physical Sky*. De este modo las nubes se teñirán de las características dadas por el *Physical Sky* sin necesidad de substituir el mapa. Dado que los rangos de intensidades de un sistema de luz diurna son muy altos habrá que modificar las intensidades de salida del mapa. El procedimiento será el siguiente:

- a) Desde el Editor de Materiales, editar los parámetros de *mr Physical Sky* y desactivar la opción “Inherit from mr Sky.
- b) Asignar a *Haze* un mapa de bits que represente un cielo con nubes.
- c) Editar el mapa y, en *Output*, modificar la intensidad (*Output Amount*) que en principio es 1.0 a un valor en torno a 8.0.

Las figuras adjuntas muestran ejemplos del primer grupo de propiedades, las que afectan al tamaño y la intensidad del sol. Y un ejemplo de como combinar un mapa de bits con nubes con el *Physical Sky*.

Naturalmente también se puede prescindir por completo del mapa *mr physical* y utilizar cualquier otro tipo de mapa, un mapa de bits o un mapa procedural que, por ejemplo, simule un degradado configurado libremente. pero como decía, esto entra de lleno en los procedimientos de configuración de materiales y mapas que se analizan en el libro citado, incluyendo los procedimientos que hay que seguir si se quiere controlar de modo independiente los mapas de fondo que se aplican a la pantalla y los mapas de fondo que se reflejan en objetos de la escena.

Y también es posible asignar un mapa a una *skylight*. Esto se hace, principalmente, para hacer que la distribución de la luz no sea uniforme sino que varíe en función de las características del mapa.

Para que este método sea realmente eficaz es preferible utilizar un mapa de tipo HDR y las técnicas que utilizan mapas HDR para simular la iluminación global de la escena caen de lleno en los métodos de cálculo IBL (*Image Based Lighting*), que he introducido desde un punto de vista teórico en el capítulo 3 y cuyas técnicas de aplicación veremos con más detalle en el capítulo 6.

### **Sky portals**

Los *sky portals* se introdujeron inicialmente (en la versión 2008 de Max) como un recurso específico ligado a los cálculos de iluminación global por medio de fotones. Y en casos en los que la luz principal estaba situada en el exterior, y la escena que se quería representar consistía en un interior con una ventana. Esta luz exterior emitía una gran cantidad de fotones que se desperdiciaban pues sólo una pequeña fracción entraba en el interior. El *sky portal* actuaba como una especie de “embudo” que enfocaba los fotones en la dirección correcta. Esta resultó ser una opción más efectiva que utilizar otro recurso habitual, el complementar la escena con una luz de área adaptada a la ventana. Los resultados son más rápidos, la distribución de la luz es más homogénea y la iluminación es más fácil de ajustar pues el *sky portal* está preparado para adaptarse a la *skylight* del sistema de luz diurna.

Un *sky portal* se crea como un rectángulo, pinchando y arrastrando. Si se quiere alinear con una ventana lo más recomendable es hacerlo desde el exterior (para que apunte al interior) y en modo *autogrid* (para que se adapte al plano del muro sobre el que se va a colocar). No hay que hacer mucho más pues los parámetros por defecto suelen funcionar correctamente. Los parámetros principales son los siguientes:



El *Multiplier* (1.0) permite ajustar la intensidad, igual que en las demás luces. *Filter Color* (blanco) permite modificar el color.

El grupo *Shadows* permite activar o desactivar las sombras. La casilla "From Outdoors" (desactivada) se puede activar si interesa que objetos exteriores arrojen sombras. *Shadow samples* sirve como en otros casos para refinar el resultado si aparecen defectos: aumentar el valor predeterminado (16) mejora la ca-

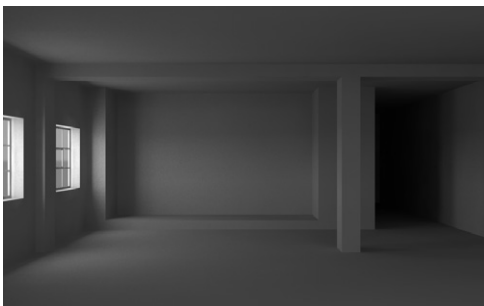
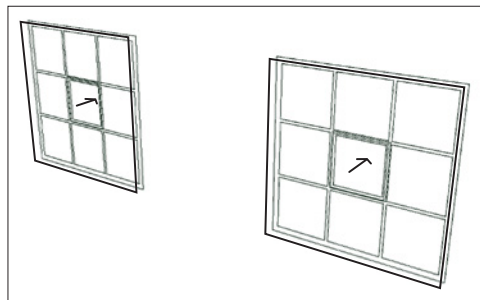
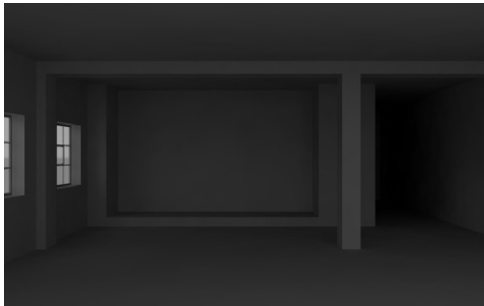


Figura 5.31 Sky Portals: a) Imagen obtenida sin los portals y un sistema de luz diurna con solo la Skylight activa y el control de exposición en EV 14; b) Esquema de la colocación de los portals sobre las ventanas, vistos desde el exterior; c) Imagen obtenida con los portals activados y el multiplicador en 10.0

lidad a costa de más tiempo de computación. Desde el grupo *Dimensions* pueden ajustarse la anchura y longitud del portal. Si la flecha que indica la dirección en que se emite la luz no apuntara en la dirección correcta se puede invertir con *Flip Light Flux Direction*.

La sección *Advanced Parameters* incluye parámetros adicionales. *Visible to Renderer* (desactivado) puede hacer que el *sky portal* se vea en la representación de salida, algo que raras veces interesa. *Transparency* sirve para modificar (oscurecer) los objetos del exterior que puedan verse a través de la ventana. El valor dado por *Colour Source* está predeterminado para derivar el color del *portal* del dado por la *Skylight* pero pueden usarse otras dos opciones: el color dado por el mapa de entorno (*Use Scene Environment*) o un color dado por un mapa escogido por el usuario (*Custom*).

Las figuras adjuntas muestran un ejemplo de como utilizar este recurso. El procedimiento básico en 3ds Max es el siguiente:

1. Crear un sistema de luz diurna siguiendo los procedimientos que ya se han explicado. Desactivar el sol. Para una escena como la del ejemplo, ajustar la configuración de cálculo a *FG Medium*, con 2 rebotes y el Control de Exposición con EV 14.

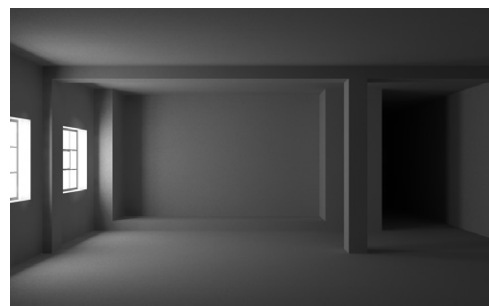


Figura 5.32 Sky Portals. Imagen obtenida sin sistema de luz diurna y los portals funcionando exclusivamente como luces extensas con el multiplicador en 100 y el control de exposición en EV 3.5



- Con esta configuración se obtendrá un resultado como el de la figura adjunta (a).
2. En el panel *Create/Lights*, seleccionar, de la lista de luces fotométricas, *mr Sky Portal*. Situarlo en el exterior del edificio, activar *Autogrid* para que el *portal* se alinee con la cara sobre la que se sitúe el cursor y pinchar y arrastrar sobre el muro para crear un rectángulo que representará el portal. Luego alinearlos con una de las ventanas, cambiar sus dimensiones para que sean iguales o algo mayores que el hueco y desplazarlo ligeramente hacia fuera. Clonar este portal como instancia (para que los cambios de uno se transmitan automáticamente al otro y alinearlos con la segunda ventana. La figura adjunta (b) muestra un esquema de los dos *portals* colocados sobre las ventanas de este modelo.
  3. Activar las sombras del *mr Sky Portal*, comprobar que en parámetros avanzados está seleccionada la opción "Use existing Skylight" y repetir el *render*.

El resultado mostrará una distribución mejor de las luces en el interior además de sombras difusas proyectadas desde el hueco. El efecto es sutil pero puede aumentarse jugando con el multiplicador del portal lo que es una ventaja adicional.

4. Editar el portal. Cambiar el valor del multiplicador (1.0) por 10.0 y aumentar el número de muestras de las sombras hasta 64 para reducir el ruido que aparecerá con este tipo de cálculo. Repetir el *render*. El resultado deberá ser similar al de la figura adjunta (c).

Aparte de su uso corriente para redirigir la luz de *skylights* a interiores, puede ser utilizado como alternativa a luces fotométricas de área pues da sombras suaves mejor distribuidas.

Para comprobarlo, continuar con el ejemplo anterior como sigue:

1. Eliminar o desactivar el sistema de luz diurna.
2. Editar el *sky portal*. En parámetros avanzados, cambiar la opción "Use existing Skylight" por "Custom". Aumentar el valor del multiplicador a 100.
3. Cambiar el Control de Exposición reduciéndolo a la cuarta parte aproximadamente. En el ejemplo se ha fijado en EV 3.5.
4. Repetir el *render*. El resultado deberá ser similar al de la siguiente figura adjunta.

Si el *sky portal* se coloca muy cerca de una abertura puede hacer que los bordes cercanos queden excesivamente iluminados. Para solucionar este problema utilizar alguno de estos recursos o una combinación de ellos: a) desplazar el portal alejándolo un poco de la abertura, b) reducir su tamaño y reajustar la intensidad (la intensidad es proporcional al tamaño del *sky portal*), c) rebajar el valor del multiplicador, d) en el Control de Exposición, rebajar el valor de *Highlights (burn)* de 0.25 a 0.05 o menos.

También es posible que aparezca ruido de diversos tipos. Una causa elemental de ruido, que ya hemos visto, es el insuficiente número de muestras para el cálculo de sombras suaves. Esto se soluciona editando el *Sky Portal* y aumentando el valor de *Shadow Samples* que, en principio, es 16. Descolgar la lista que muestra este valor y escoger valores superiores: 32, 64, 128... (potencias de 2 en todos los casos).

También es posible que aparezca ruido debido a la influencia del disco solar que, aunque no sea visible en la escena, puede afectar a las superficies. Si aparecieran puntos blancos o puntos irregulares en torno a las sombras una solución que a menudo elimina el problema directamente es editar el *mr Physical Sky* y, en el grupo *Sun Appearance*, cambiar los valores predeterminados para *Disk Intensity* (1.0), *Glow Intensity* (1.0) y *Scale* (4.0) a 0.0, 0.0 y 1.0 respectivamente.



## Control de la temperatura de color en *sky portals*

Al utilizar un *Sky Portal* como luz independiente el resultado inicial será gris pues la fuente de color ha dejado de ser la *skylight* o un mapa de entorno.

Pero se puede controlar directamente el color de un *Sky Portal* por medio de un recurso especial, un *shader* que afecta al color de la luz. Este *shader* (que estaba oculto en versiones anteriores a la 2012) se denomina *Kelvin Temperature Color*.

Todo lo que hay que hacer es pulsar el botón junto a la opción *Custom* y escoger un mapa de *mental ray* tipo *Kelvin Temperature Color*.

Luego arrastrarlo al Editor de Materiales para editarlo. Solo incluye dos parámetros: *Kelvin Temperature* (6.500°K predeterminado) e *Intensity* (1.0).

El valor predeterminado, 6500°K da una tonalidad neutra. Para modificarlo, utilizar valores en torno a 3500°K para tonos cálidos o en torno a 7500°K para tonos fríos.

Si se aumentan o disminuyen los valores de intensidad se modificarán los tonos pero esto también afectarán a la iluminación global de la escena por lo que, en general, es preferible no tocar el parámetro que afecta a la intensidad y limitarse a utilizar este recurso para modificar la temperatura de color.

## 5.5 Luminarias y objetos luminosos

### Luminarias

Las luces que hemos visto hasta ahora emiten luz pero no son visibles: son puntos o formas en el espacio que actúan sobre la escena pero sólo se ven sus efectos.

En los escenarios reales esto no es así, como todos sabemos: vemos la luz pero también podemos ver el objeto que produce la luz. Sin embargo, en los escenarios virtuales

la “luz” no es sino una especificación numérica que hace que los objetos se procesen de un modo determinado.

Como veremos en los siguientes apartados, también es posible crear objetos que emitan luz. Sin embargo la luz emitida por estos objetos no se puede procesar con la precisión adecuada excepto con algunos métodos de cálculo y, aún así, con limitaciones importantes. Por esta razón, en la gran mayoría de los casos hay que recurrir a un procedimiento bastante artificioso: crear la luz y, además, crear el objeto que simula la luz y asociarlo a la luz haciendo a menudo, por otra parte, que la luz no lo tenga en cuenta, que lo excluya de sus cálculos, para que no haya interferencias tales como que la luz no pueda iluminar nada porque está dentro del objeto o que el objeto luminoso arroje sombras indeseadas.

Las posibilidades de este tipo de asociaciones son infinitas. El libro de Simulación de Materiales presenta ejemplos más elaborados de cómo llevar esto a cabo. Lo que sigue son tres ejemplos muy elementales que resumen estas posibilidades para situar este tipo de simulación en el marco de un capítulo que trata de luces y sombras.

### Tres ejemplos

El primer ejemplo muestra un objeto con una serie de elementos que forman la lámpara. Y, en su interior, dos luces *spot* que apuntan en direcciones opuestas y simulan un efecto de difusión. Para crearlas el procedimiento que hay que seguir, y que se resumen en el esquema de la figura adjunta es:

1. Crear dos luces *standard* de tipo *spot* en la posición de la bombilla. Primero crear una, de tal modo que su *hotspot* quede abierto como en el esquema de la figura y el *fall-off* más abierto y rozando los bordes de la pantalla. Luego clonarla como instancia y mover su *target* en dirección opuesta para crear la segunda.
2. Seleccionar una de las dos luces, editarla y en la sección *General* presionar el botón



- Exclude.* Seleccionar todos los objetos que conforman la lámpara y excluirlos (o si se prefiere, excluir sólo la bombilla y la pantalla y los elementos internos de la pantalla).
3. Asignar al elemento que simula la pantalla un material autoiluminante. Para ello, abrir el Editor de Materiales y crear y probar algunos de los tres siguientes: a) Material de tipo *Standard*, de color blanco cremoso y autoiluminación al 100%. O bien, alternativamente, un material de tipo *Arch&Design* con la sección *Self-Illumination* activada y los parámetros predeterminados, b) El mismo material pero con un mapa asignado a la propiedad de autoiluminación. El mapa sería de tipo *Gradient Ramp*, girado 90° en W para que el degradado sea vertical y configurado de tal modo que sea blanco en zona central y gris suave en los extremos.

La parte blanca deja actuar plenamente la autoiluminación. La parte gris la atenúa; c) El mismo material pero con un mapa que simule la textura de la pantalla y con la textura trabajada en un programa de pintura digital para que sea más luminosa en la parte central. Ver el libro sobre Simulación de Materiales para ejemplos y descripciones más elaboradas de simulación de materiales autoiluminantes.

El segundo ejemplo muestra un objeto similar, en este caso con una sola luz *standard spot* pero que también está configurada de modo que simule un efecto de difusión. El resultado, como se aprecia en la figura adjunta (a) es que la pantalla parece irradiar como si se tratara de un objeto extenso, creando zonas de penumbra. El material que se ha asignado en este caso a la luminaria es un material de tipo *Top/Bottom* que resulta cómodo para un caso como este pues permite utilizar dos materiales que se aplican según la orientación de la normal a la superficie. Como en este caso la carcasa y su interior apuntan claramente en dos direcciones opuestas, todo lo que hay que hacer es elegir un material opaco y oscuro para la carcasa (*Top*) y un material autoiluminante para el interior (*Bottom*). Por añadidura (b) se le ha añadido un efecto de volumen de luz, que se crea con facilidad desde la sección *Atmosphere and Effects* del panel de edición de las propiedades de la luz y al que volveremos más adelante.

El tercer ejemplo se basa en objetos simples, cilindros con su parte inferior autoiluminante. Puede utilizarse un recurso similar al del ejemplo anterior, un material *Top/Bottom* o, si esto no funcionase (porque el giro de la normal es demasiado brusco) un material multisubobjeto o, más sencillamente, separar el disco inferior y aplicarle un material autoiluminante. En este caso se han utilizado luces fotométricas con los IES semifocales genéricos que he descrito en el apartado correspondiente más arriba.

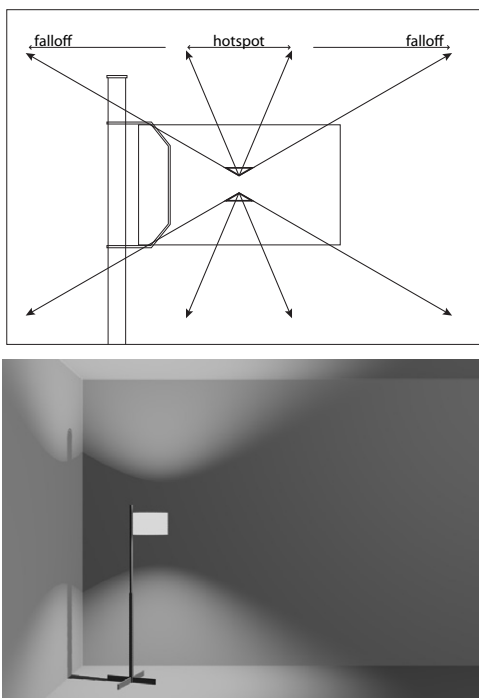


Figura 5.33 Ejemplo 1. Luminaria creada agrupando objetos con autoiluminación y luces standard: a) esquema de la disposición de las luces, b) utilización en un recinto simple

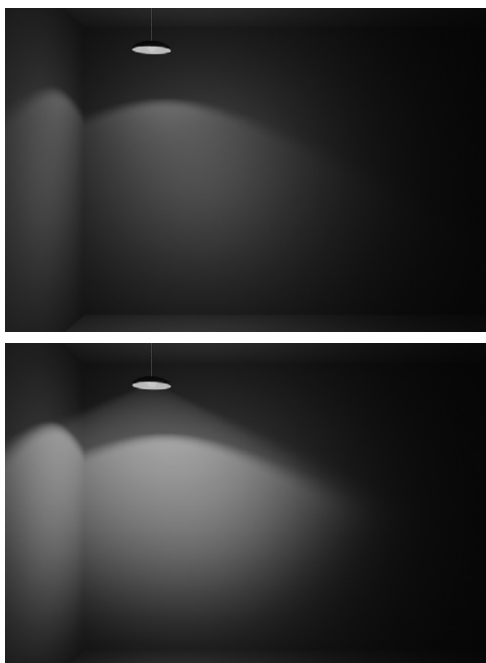


Figura 5.34 Ejemplo 2. Luminaria creada de modo similar al ejemplo anterior y con adición de volumen de luz

Tener en cuenta que cuando se utilizan luces próximas a objetos, las áreas cercanas pueden aparecer quemadas debido a un exceso de intensidad, principalmente cuando se usan fotones como apoyo al cálculo de FG en luces situadas dentro de luminarias (ver el siguiente capítulo). Si sucede algo así probar alguna de estas correcciones: a) Mover las luces fotométricas alejándolas un poco de la luminaria; b) Cambiar la distribución de la luz (*uniform diffuse* o *spot*) para dirigir los fotones lejos de la luminaria; c) Cambiar las propiedades del interior de la luminaria para que no reciba fotones.

Por último, si, como acabamos de ver, es habitual asociar luces a objetos autoiluminantes, también resulta conveniente agrupar los objetos autoiluminantes y las luces para poder moverlos con facilidad en caso de hacer cambios en la escena. Pero si se quiere mo-

dificar la intensidad o el color de la luz, habrá que desagruparlos, modificar la intensidad o cualquier otro parámetro y, luego, volverlos a agruparlos, lo que resulta incómodo.

Esta es la razón de ser de los *ensamblajes* (*assemblies*), un tipo de objeto o recurso, disponible en 3ds Max a partir de la versión 4.0, que permite combinar una serie de serie de elementos geométricos y una o varias luces de modo que funcionen como un único objeto. La diferencia principal con respecto a un grupo es que se añade automáticamente un objeto auxiliar, denominado *Luminaria*, que permite controlar la intensidad y el color de la luz sin necesidad de desagrupar los componentes. Por lo demás funcionan igual que los grupos y con la misma convención de mostrar los nombres entre corchetes y en negrita. Pueden también crearse instancias, de modo que las modificaciones se transmitan automáticamente. Sin embargo, a diferencia de los grupos, la aplicación de un modificador sólo afecta al objeto ayudante, no a los componentes.

Sin embargo es un procedimiento más bien tedioso y poco utilizado en la práctica desde que apareció otro recurso más efectivo, por lo que he eliminado su descripción en este apartado ya que solo puede tener sentido en casos muy especiales. Puede encontrarse una descripción más amplia en la Ayuda de Max.

El otro recurso, más efectivo y suficiente para la gran mayoría de los casos, es utilizar la herramienta *Light Lister*, que se encuen-

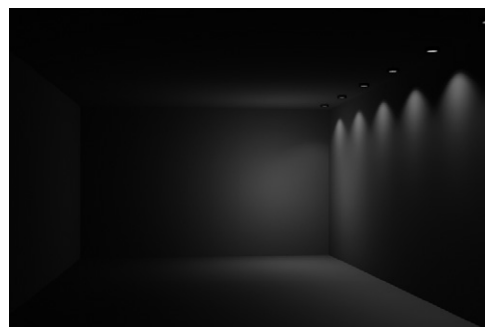


Figura 5.35 Ejemplo 3. Luminarias creadas agrupando discos autoiluminantes y luces fotométricas



tra en el menú *Tools* de 3ds Max y al que ya me he referido más arriba. Esta herramienta presenta un panel con todas las luces de la escena y es posible modificar sus propiedades principales sin necesidad de tener que seleccionarla y editarlas.

### Objetos irradiantes

Como decía más arriba, también es posible crear objetos que parece que emitan luz o que emitan luz hasta cierto punto. La eficacia de

este método depende directamente del sistema de cálculo. Si se utilizan sistemas basados en *Path Tracing*, la irradiación de los objetos puede computarse directamente y no es necesario utilizar luces virtuales. Hay que tener en cuenta, no obstante, que el control de la distribución de la luz no será tan efectivo como cuando se utilizan luces que cuentan con recursos específicos para controlar la dirección y la distribución.

En programas como Maxwell, que utiliza una variante de *path tracing bidireccional* con

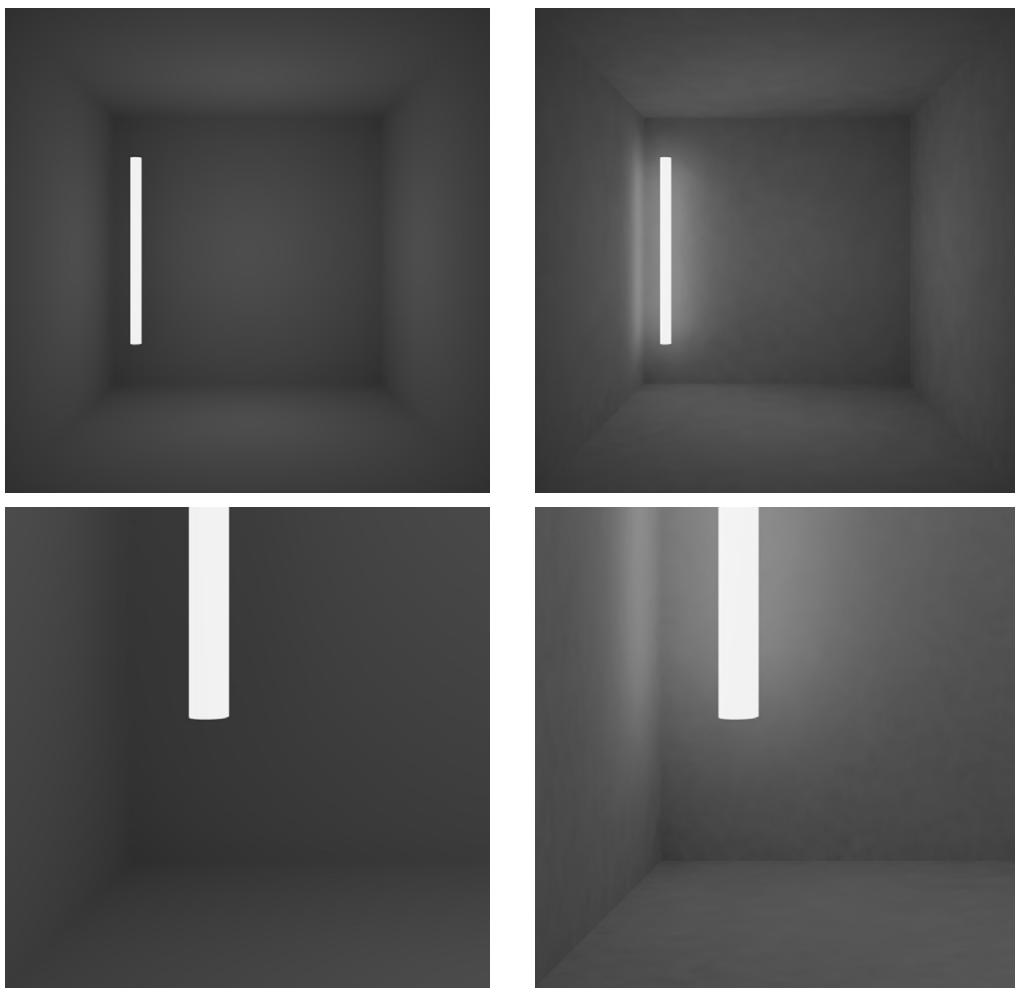


Figura 5.36 Escena con un objeto irradiante: a) Sin contribución a la iluminación, b) Con contribución a la iluminación y una intensidad de 1500 cd, c) Detalle de la primera imagen, d) Detalle de la segunda imagen



*Metropolis Light Transport* como sistema de cálculo, las luces se crean mediante objetos a los que se asignan materiales emisores. Las propiedades de estos materiales permiten especificar la intensidad de la emisión. El inconveniente, con todo, sigue siendo el citado en el párrafo anterior: no se puede controlar suficientemente la distribución de la luz lo que puede suponer una limitación importante en muchos casos.

En las últimas versiones de los programas que utilizan mental ray (3ds Max, Maya, Soft Image) se puede utilizar un nuevo motor de render denominado *iRay* que utiliza una variante de *path tracing*.

En el siguiente capítulo veremos métodos basados en estos sistemas. En este apartado me limitaré a describir como los sistemas que no utilizan estos sistemas de cálculo pueden sacar partido, hasta cierto punto a la defini-

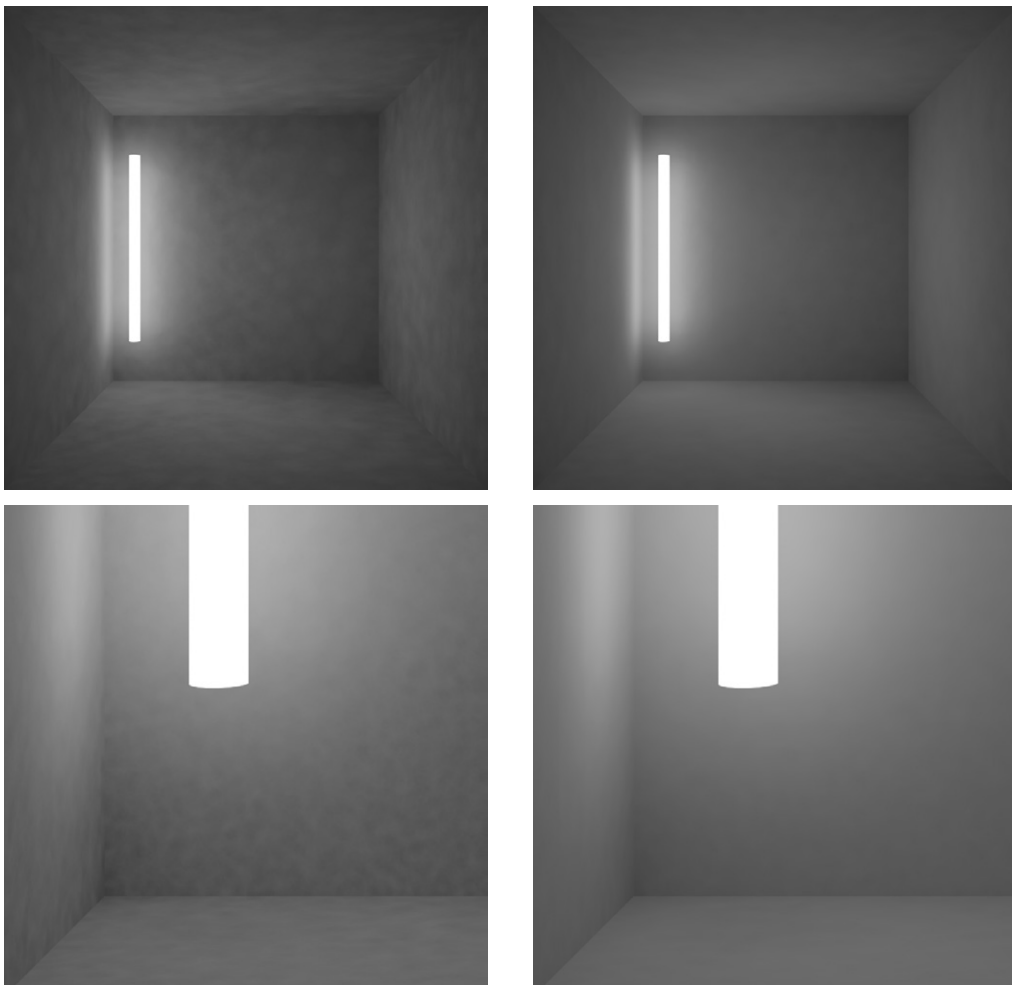


Figura 5.37 Escena con un objeto irradiante que contribuye a la iluminación con una intensidad de 3000 cd: a) Cálculo con FG media, b) Cálculo con FG alta, c) Detalle de la primera imagen, d) Detalle de la segunda imagen



ción de un objeto como “irradiante” o “emisor” de luz.

En sistemas como *Radiosity* o *Final Gather* pueden incorporarse objetos autoiluminantes pero con limitaciones importantes. Puede ser interesante para simular ciertos efectos pero no para crear un sistema de iluminación efectivo. En *mental ray* esto se hace a través de una especificación de material que ya he mencionado. Y, como también he dicho, esto se trata más extensamente en el libro sobre Simulación de Materiales. En este apartado me limitaré por tanto a una introducción que complete el panorama general de uso de luces virtuales.

Hay dos modos de usar esta especificación. Con el primero, no hay propiamente emisión (aunque el objeto autoiluminante sí afecta a los que le rodean de modo distinto). Con el segundo hay cierto grado de emisión pero no es comparable a la de una luz virtual.

Lo ilustraré con algunos ejemplos sencillos. Las figuras adjuntas muestran un recinto de 3x3x3 m con una luz fotométrica puntual en el centro (no visible) de 500 cd. El cálculo en todos los casos se ha hecho con *Final Gather* de calidad media y 2 rebotes. El control de exposición es de tipo *mr Photographic Exposure Control* ajustado a 10 EV.

En este recinto he situado un cilindro de 12 cm de radio y 2 m de altura, a 20 cm de las paredes más cercanas. A este objeto le he asignado un material *Arch&Design* con *Self-Illumination (Glow)*, activado. También se puede utilizar un material *Standard* con autoiluminación pero las posibilidades de *Arch&Design* son mayores. Se puede especificar la intensidad de la irradiación en unidades fotométricas (*physical units*, cd) o convencionales (*unitless*). Y, en la sección *Self-Illumination* incluye un parámetro, *Illuminates the scene when using FG*, que hace que contribuya, si se activa, a la iluminación de la escena.

La primera figura se ha obtenido con este parámetro desactivado.

La segunda, con la misma configuración pero el parámetro de iluminación activado y una intensidad propia de 1500 cd. El resul-

tado es que la zona adyacente recibe más iluminación pues computa este valor de irradiación.

Sin embargo, como se puede apreciar en la vista de detalle, aparecen defectos notorios.

Si aumentamos la intensidad, las imperfecciones serán aún mayores, lo que obliga a ajustar la configuración de cálculo. El siguiente grupo de figuras muestra, en primer lugar, a la izquierda, el resultado que se obtiene con una intensidad de 3000 cd y la misma configuración que en el grupo anterior y, a la derecha, el resultado que se obtiene con la configuración ajustada para reducir las imperfecciones. Los parámetros principales que se han modificado y que se explicarán con más detalle en el siguiente capítulo, son los correspondientes al cálculo de FG en los que se ha pasado de una configuración media (0.8 de densidad, 250 rayos por punto, 30 puntos de interpolación) a una configuración alta (1.5 de densidad, 500 rayos por punto, 100 puntos de interpolación). Pero el tiempo de cálculo ha sido casi 4 veces superior en este segundo caso. En la parte inferior de la figura se muestra un detalle de las dos imágenes. Como puede apreciarse (aunque en la impresión sobre papel se pierde calidad) el resultado es algo mejor pero no *mucho* mejor. La conclusión es que no deben utilizarse los objetos luminosos como fuentes de luz de la iluminación de la escena sino como complementos y, aún en este caso, debe cuidarse de los efectos que tendrán sobre otras superficies cercanas.

### Efectos de volumen de luz y de resplandor (*glow, glare*)

Si que quiere que la zona que atraviesa la luz quede resaltada, como ocurre en algunos casos cuando hay partículas en suspensión, el foco está bastante concentrado y hay suficiente contraste, hay que utilizar efectos de volumen de luz.

Este tipo de efectos también se describen más ampliamente en el libro de Simu-



lación de Materiales dado que esto entra más bien el terreno de representación de propiedades físicas de sustancias sólidas o semisólidas. Aquí me limitaré a algunas indicaciones básicas.

Este efecto, *Light Volume* (Volumen de luz), está disponible, en 3ds Max, en el cuadro de diálogo *Environment&Effects*, en la sección de *Efectos de Atmósfera*. También se puede acceder a este efecto editando la luz desde el panel *Modify* y desde la sección *Atmósferas y efectos*. Ya me he referido a esta posibilidad más arriba. Al editar el efecto quedan disponibles parámetros para cambiar el color o la densidad del volumen de luz, su velocidad de atenuación, etc. Las figuras adjuntas muestran algunos ejemplos.

Otro tanto cabe decir de efectos de resplandor que también se tratan más extensamente en ese libro. Descrito de un modo muy sucinto, hay dos tipos de técnicas disponibles en 3ds Max: las incluidas con el motor corriente, que se encuentran en el menú *Environment and Effects* y las que se encuentran en el módulo *Video Post* al que se puede acceder desde el menú *Rendering*. Las primeras no pueden utilizarse con mental ray por lo que es más recomendable utilizar *Video Post* que incluye una gama muy amplia de efectos y está disponible para los dos motores, el predeterminado de 3ds Max y el de mental ray. Como el anterior, se describe con más detalle en el libro sobre Simulación de Materiales. Las figuras adjuntas muestran algunos ejemplos.



Figura 5.38 Ejemplos de Volumen de luz

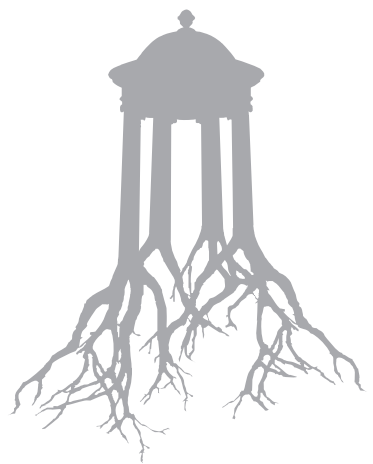


Figura 5.39 Ejemplos de efectos de resplandor





Debe tenerse en cuenta que estamos hablando de efectos de **postproducción**, es decir de efectos que **se superponen** a la imagen generada con un determinado sistema de *render*. Por consiguiente tanto da crear estos efectos desde un programa de manipulación de imágenes como Photoshop o Gimp. Lo que se gana por un lado (mayor comodidad y rapidez si se hace sin salir del programa de *rendering*) se puede perder por otro (técnicas más limitadas y menor control sobre la imagen).



→ 6



# Sistemas y configuraciones de cálculo de iluminación global

## 6.1 Iluminación directa e iluminación indirecta

### Sistemas de iluminación local y de iluminación global

En muchos programas sencillos, la simulación de la iluminación por medios informáticos se obtiene a partir de lo que se denomina actualmente “sistemas de iluminación local”, por oposición a “sistemas de iluminación global” que es el tema principal de este capítulo. En los programas avanzados también se utilizan los sistemas de iluminación local para mostrar la escena en el visor de modo que el usuario pueda interactuar más fácilmente con el modelo.

En estos sistemas, las luces se relacionan con los objetos como mónadas aisladas, sin tener en cuenta la interacción entre ellos. Esto explica lo artificioso de muchas de las técnicas que buscan compensar estas limitaciones. Como ya hemos visto en capítulos anteriores, el cálculo primario de la iluminación se hace por medio de una multiplicación vectorial de los dos vectores unificados que representan la dirección de la luz y la normal a la superficie. Este cálculo es muy sencillo y está incorporado a todas las tarjetas gráficas actuales por lo que se realiza con una enorme rapidez y permite visualizar fácilmente los objetos. Pero esto es todo. Y, las limitaciones, que también hemos visto en el capítulo anterior, son muy importantes.

En los sistemas de iluminación global o iluminación avanzada, estas limitaciones se corrigen mediante procedimientos mucho más complejos. Pero en este caso aparecen otro tipo de limitaciones, debido precisamente a la complejidad del cálculo.

Antes de entrar en la descripción técnica de las configuraciones adecuadas para este

tipo de sistemas, conviene aclarar que ambos tienen muchas cosas en común y que, según la aplicación empleada, la frontera entre ambos métodos varía.

Podemos sintetizar las características más o menos comunes en los siguientes grupos de categorías, que se resumen en los siguientes apartados: la disposición de las luces, la geometría de la escena, las características de los materiales, el método de cálculo de las sombras, la dependencia del punto de vista y de la resolución y los métodos de *antialiasing*.

Además de esto, es importante no perder de vista que la mayoría de los sistemas de iluminación avanzada separan el proceso de cálculo en dos grupos: el cálculo de la luz directa y el cálculo de la iluminación indirecta.

### Tipos de organización de la escena y tipos de luces en ambos sistemas

En primer lugar, las características de un escenario virtual dependerán, obviamente, de la iluminación, de las fuentes de luz presentes en la escena. El modo de colocar las luces en la escena es un problema de organización y de diseño, y no depende de la técnica de simulación que vayamos a emplear. Pero es un problema conceptual al que se debe conceder toda la importancia que tiene.

Ahora bien, suponiendo que esta elección ya se haya llevado a cabo, la **distribución inicial de la iluminación** será la misma, tanto si utilizamos un sistema de iluminación local como global. Los sistemas de iluminación global parten de una condición inicial que es la misma que la de los sistemas locales y, a partir de ahí, se desarrollan en diversos estadios en los que se redistribuye el estadio inicial.

Esto quiere decir, para continuar, que la apariencia inicial, la distribución fijada en este primer estadio, depende de la **geometría**



**de la escena.** Si consideráramos la escena como una formada por una serie de superficies planas compuestas de diferentes zonas, la intensidad inicial de estas zonas dependería, si suponemos que sus propiedades materiales no son relevantes, exclusivamente de su orientación con respecto a la luz. El problema conceptual se amplía pues esta apariencia inicial es determinante para la imagen que queremos conseguir.

Pero, en tercer lugar, las **propiedades materiales** sí son relevantes. Por lo que también habrá que tener en cuenta la mayor o menor claridad de estas superficies que harán que la luz se refleje con mayor o menor intensidad aunque no computemos los reflejos entre superficies.

Antes de continuar (en los siguientes apartados) volvamos al comienzo, a la distribución inicial de la iluminación que depende de las luces que hayamos situado en la escena.

Lo principal sigue siendo común. Y lo principal es la posición y la orientación de las luces. También es importante el tamaño, la extensión de estas luces. Ya hemos visto que los sistemas de iluminación simple permiten utilizar luces extensas por lo que, con la salvedad de que los diferentes tipos de luces llevan incorporada esta propiedad de modos diferentes, tampoco aquí encontramos diferencias de importancia.

Sin embargo sí hay una diferencia técnica que aparecerá en los siguientes capítulos. Los diferentes sistemas de iluminación global difieren con respecto al tipo de luces que reconocen. Algunos sólo reconocen tipos de luces fotométricas. Otros reconocen tanto luces fotométricas como luces simples o luces “estándar”. Y hay otros que reconocen como fuentes de luz a objetos con propiedades irradiantes.

Entre los primeros están los sistemas de radiosidad y algunos sistemas de *path tracing*. Entre los segundos están la mayoría de sistemas de irradiancia y de proyección de fotones. Estas diferencias deberán tenerse en cuenta en la práctica y las iremos viendo a medida que describamos los diferentes métodos de cálculo.

## Tipos de cálculo de sombras en ambos sistemas

Desde el punto de vista que nos ocupa, podemos decir que habría dos grandes grupos de sistemas: los que efectúan un cálculo independiente de las sombras y lo componen con el cálculo general de la iluminación y los que integran el cálculo de sombras en el de iluminación.

Todos los sistemas de iluminación local pertenecen al primer grupo. Pero, en el caso, de los sistemas de iluminación global de nuevo nos encontramos con variantes. Algunos sistemas funcionan como los locales, esto es, superponen el cálculo de sombras al de iluminación, otros no reconocen los sistemas de cálculo de sombras incorporados a las luces, del tipo que sean y otros permiten integrar o no este cálculo en el resultado final.

Los sistemas de irradiancia y proyección de fotones computan las sombras por medio de un cálculo paralelo (lo que hace también que el resultado sea más rápido) y lo integran en el cálculo final. Los sistemas de *path tracing*, por lo general, no tienen en cuenta este cálculo y las sombras son un resultado natural del proceso de cálculo global. Los sistemas de radiosidad permiten elegir entre combinar el cálculo de sombras propio de las luces con en el resultado final, lo que da sombras más precisas o integrar este cálculo en el procedimiento general, lo que da sombras más difusas.

## Dependencia del punto de vista y de la resolución. Métodos de filtrado y *anti-aliasing*

La calidad de la imagen final depende de varios parámetros que se pueden ajustar para obtener resultados de menor calidad en menos tiempo o, viceversa, de mayor calidad, a costa de mayor tiempo de computación. Como es lógico, lo más corriente es empezar por la primera opción, menos calidad en menos tiempo, para hacer pruebas y terminar con la última, mayor calidad en mayor tiempo.



Estos parámetros son de varios tipos. Unos son independientes hasta cierto punto del sistema de cálculo y otros dependen específicamente del sistema de cálculo.

Los parámetros que son independientes del sistema de cálculo y son por tanto comunes a los sistemas de iluminación local o global son los que tienen que ver con la resolución y el *antialiasing*. Los dependientes de cada sistema los iremos viendo en cada caso.

La calidad de la imagen depende en primer lugar de la resolución. Con una resolución muy baja, es decir, con una imagen de salida pequeña, muchos detalles desaparecerán por la sencilla razón de que caen fuera de la trama de muestras tomadas por el sistema para computar la imagen. Peor aún, esta pérdida es aleatoria pues depende de si un determinado punto de un mismo objeto, cae en el centro del sistema de captación y es convenientemente registrado o cae entre dos centros con lo que no es registrado. Cualquier pequeño movimiento de la cámara puede variar esta situación y hacer que un punto pase a ser visible o invisible.

Tanto en un sistema local como global, este problema puede obviarse al comienzo pues es conveniente hacer pruebas a pequeño tamaño para comprobar si la iluminación básica es la que interesa y luego aumentar el tamaño a una resolución algo mayor o hacer *renders* parciales de determinadas zonas para, finalmente, cambiar el tamaño al necesario para el destino final de la imagen. Pero conviene recordar que, en estas pruebas, hay ciertos detalles que, sencillamente, *desaparecen* (por ejemplo, juntas finas en un paramento) lo que puede desconcertar a un usuario poco experimentado.

Y la calidad de la imagen depende en segundo lugar del sistema de *antialiasing* utilizado. Los sistemas básicos ya se han resumido en el capítulo 3. Muchos de estos sistemas son, como he dicho, independientes del sistema de cálculo: una línea inclinada aparecerá más o menos escalonada según estemos forzando al sistema a tomar más o menos muestras globales lo que resulta equivalen-

te a aumentar virtualmente la resolución por métodos de supermuestreo (*supersampling*) y luego tomar valores intermedios para evitar saltos bruscos.

### **Ventajas de los sistemas de iluminación local**

Pero tampoco deben perderse de vista las ventajas de los sistemas de iluminación local. Las luces simples son sencillas de comprender y de utilizar, son rápidas de calcular, son flexibles y se manejan con facilidad. Por esta razón se siguen utilizando con dos finalidades principales: para crear escenas simples en las que el realismo no es la finalidad principal y para complementar escenas realistas por medio de luces secundarias que iluminen determinadas zonas con facilidad sin recargar el cálculo y la disposición general de las luces principales.

Muchos especialistas en simulación, que llevan años trabajando con luces estándar, las siguen utilizando para conseguir efectos de iluminación avanzada. Pueden simularse luces extensas de un modo bastante efectivo por medio de colecciones de luces estándar, dispuestas a intervalos regulares sobre un plano virtual o una esfera o un cilindro. La atenuación puede especificarse mediante los parámetros propios de las luces estándar, como ya hemos visto. Y el efecto de rebote por medio de luces secundarias. Aunque la preparación de un escenario de estas características resultará bastante pesada para alguien que se inicia en técnicas de simulación, la velocidad del cálculo será bastante mayor y los resultados pueden ser comparables a los de sistemas de iluminación global.

### **Ventajas de los sistemas de iluminación global y sumario de los descritos en este capítulo**

Las ventajas de los sistemas de iluminación global son las opuestas. Proporcionan mayor realismo a costa de mayores complicaciones en la preparación y configuración de la esce-





na y, sobre todo, a costa de mucho más tiempo de cálculo que, en determinados casos, puede ser del orden de varias horas.

La razón de ser de la mayoría de los sistemas de iluminación avanzada viene, precisamente, de los intentos de compensar estos inconvenientes, más concretamente, de reducir el tiempo de cálculo sin pérdida significativa de precisión. Naturalmente, lo que queremos decir con “pérdida significativa” es algo que no tiene respuesta si no sabemos cuál es la finalidad del proyecto en que estamos metidos.

Por estas razones, la descripción de las ventajas y desventajas de los diferentes sistemas, que ya se ha dado en el capítulo 3, sólo puede repetirse aquí, de modo más condensado, y apuntando a algunos de los principales programas comerciales disponibles en la segunda década del siglo XXI y que son los que utilizaré en este capítulo. El lector deberá juzgar, a partir de este marco general que es lo que le puede interesar.

Hay que comenzar por distinguir entre motores de render (*render engines*) y aplicaciones. Las aplicaciones más conocidas y utilizadas, como *Maya*, *3ds Max*, *Soft Image*, *Lightwave*, *Cinema 4D* o *Blender*, proporcionan un interface complejo y sofisticado, una gran cantidad de recursos de todo tipo y un sistema de cálculo de la iluminación que se basa en un motor de *render* externo. Este motor externo, en algunos casos está incorporado a la aplicación anfitriona mediante algún tipo de acuerdo o, en otros casos, puede instalarse como plugin.

Por ejemplo, los tres primeros programas citados, *Maya*, *3ds Max* y *Soft Image*, utilizan *mental ray* como motor principal incorporado al programa que, a su vez, utiliza diferentes sistemas de cálculo que veremos más abajo. Por otro lado, el mismo *3ds Max* incorpora, desde sus inicios, otro motor propio que trabaja con Radiosidad.

En otros casos el motor de render se instala como plugin. Este es el caso de, por ejemplo, *V-Ray* que puede instalarse sobre diferentes plataformas: *3ds Max*, *Maya*, *SoftImage*, *Rhino*, *Sketchup* o *Blender* y que, has-

ta hace poco no funcionaba como programa independiente. Otros, como *Maxwell*, dan la posibilidad, desde sus inicios, de utilizarse de ambos modos.

Hay también plataformas que incluyen métodos propios, como *RenderMan* o variantes particulares de otros métodos, como ocurre con el propio *RenderMan* o *Cinema 4D* o *Lightwave*.

Un programa de aplicación que también debe mencionarse por, entre otras cosas, ser una de las pocas aplicaciones gratuitas de que se puede disponer en la actualidad es *Blender*. El motor básico de *Blender* incorpora un módulo de *ray tracing*. Y, desde la versión 2.28, un módulo de radiosidad que no necesita luces virtuales sino que puede funcionar con objetos emisores. En las versiones recientes se ha incorporado una conexión, con *YafRay* (*Yet Another Free Raytracer*, desarrollado principalmente por Alejandro Conty Estévez y Alfredo de Greef, entre otros), un motor también gratuito, basado en *ray tracing*, que usa un programa de descripción de escenas en formato XML. Esto implica dos modos de utilización: una relativamente directa, con *YafRaf* como *plugin* y otra exportando la escena de *Blender* al programa en formato XML. Puede utilizar también IBL, con archivos HDR y fotones lo que permite, entre otras cosas, calcular cáusticas. Desde la versión 2.61 también incluye otro motor de *render*, *Cycles Render Engine*, un motor basado en *ray tracing* que incluye *render* interactivo basado en aceleración por GPU por medio de dos modos, uno basado en CUDA, para tarjetas Nvidia y otro basado en OpenCL para tarjetas AMD/ATI. Aún se encuentra en fase de desarrollo más bien experimental pero incorpora cosas tales como *Ambient Occlusion*.

Hay otros programas que también cabe mencionar. Un programa popular y muy sencillo de utilizar, como *Artlantis*, utiliza una versión simplificada y optimizada de radiosidad que funciona suficientemente bien en exteriores, aunque no tanto en interiores.

Y, en fin, no debe perderse de vista que están apareciendo nuevos programas, más



sencillos, que buscan sacar el máximo rendimiento de las tarjetas gráficas y que pueden llevar a modificar el panorama, dominado por los programas citados, en breve plazo.

En los apartados que sigue resumiré, en primer lugar, los modos en que cabría ordenar los principales motores de *render* y, a continuación, una breve descripción, en orden cronológico, de los principales motores. No debe verse este resumen como una (imposible) relación exhaustiva sino como un esbozo del panorama, muy dinámico, con que nos encontramos en la segunda década del siglo XXI.

### Principales aplicaciones (motores de *render*) disponibles en 2014

El sector de los motores de *render* se puede considerar dividido en los siguientes grupos:

- a) Aplicaciones completas e independientes. El ejemplo principal es *RenderMan*.
- b) Aplicaciones anfitrionas que proporcionan un interface y una gran colección de recursos pero que se basan en motores externos para el cálculo. Este es el caso de todas las más conocidas: *3ds Max*, *Maya*, *Soft Image*, *Lightwave* o *Cinema 4D*. Algunas de ellas tienen sus motores propios pero con el tiempo los han ido relegando para confiar más en los ajenos, como ha ocurrido con *3ds Max* que tiene sus propios sistemas de cálculo de GI pero en las últimas versiones utiliza prioritariamente *mental ray*.
- c) Aplicaciones que dependen por completo o casi por completo de otras aplicaciones. Este es el caso de *Mantra* (*Houdini*), *V-Ray* o *Final Render*. Y, hasta cierto punto de otras, como *Maxwell*, que se utiliza de modo independiente pero, con más frecuencia, a través de una aplicación anfitriona. La clasificación, en este caso, sería más compleja pues podríamos distinguir entre aplicaciones que están ligadas casi exclusivamente a una determinada aplicación, otras mucho más flexibles y otras que

tienden a ser independientes aunque no lo consiguen (o no les acaba de interesar).

En esta sección me centraré en el último grupo con alguna excepción, como *RenderMan* (aunque podría considerarse también como una aplicación que depende de sí misma y en este caso estaría también en el tercer grupo).

Dicho esto, y tomando en consideración los motores más utilizados, aunque es posible que me haya dejado alguno importante, por lo que me excuso, el resumen que doy a continuación está ordenado cronológicamente, a partir de la fecha de aparición del programa para evitar otros modos de ordenación más problemáticos que indico brevemente.

También se podrían ordenar por el número de usuarios que utilizan la aplicación. Este criterio favorecería a las grandes empresas y, con este criterio, el más utilizado sería sin duda *mental ray* que es utilizado por *Maya*, *3ds Max* y *Soft Image*, que pertenecen a la mayor compañía de *software* de este tipo de aplicaciones, *Autodesk*, y por la principal productora de tarjetas gráficas *Nvidia*. No debe perderse de vista que, por esta misma razón, es un programa no muy popular entre quienes intentan hacerse un sitio en el mercado o menospreciado por quienes se dirigen a grandes empresas con exigencias más estrictas, algo que es perfectamente comprensible pero que puede originar errores de juicio si se da excesivo crédito a algunos foros de Internet. Si lo he elegido como *software* principal para los ejemplos de este libro es porque me consta que es el motor más utilizado por la mayoría de los estudiantes y profesionales a los que va dirigido y que tiene varias ventajas importantes, principalmente en términos de flexibilidad y de calidad suficiente.

Otro criterio de ordenación podría etiquetarse con algo así como “prestigio consolidado”. Desde este punto de vista no creo que quepa duda de que el primer puesto le corresponda a *RenderMan* que, desde su aparición, hace ya 25 años, ha sido una referencia fundamental para todos los avances que se han producido en este campo, comen-



zando por el propio concepto de *shader*. Pero tampoco debe olvidarse que, por las mismas fechas (1983) se fundaron *Alias Research* y *Wavefront Technologies* (1984), los dos mejores productos de simulación durante muchos años y que, en 1995, fueron adquiridas por *Silicon Graphics* y se fusionaron en un único producto que, a partir de 1998 dio lugar a *Maya*. En 2003 la compañía *Alias/Wavefront* pasó a denominarse *Alias*. En 2005 *Maya* fue adquirido por *Autodesk* y en 2006 también se completó la absorción de *Alias*. Por último, también caben pocas dudas de que una aplicación como *3ds Max*, en muchos aspectos muy similar a *Maya*, es una de las más utilizadas del mundo desde su aparición en 1990 como un programa de simulación dirigido específicamente a usuarios de PC, en todo tipo de aplicaciones profesionales que pudieran llevarse a cabo con un ordenador personal. Para este sector de profesionales puede decirse que su prestigio también está suficientemente consolidado.

Otro criterio bastante más elusivo, es el juicio de expertos sobre sus características técnicas. Pero aquí las cosas se complican pues algunos expertos consideran que un criterio importante es la ponderación entre facilidad de uso y eficacia técnica y, con este criterio, parece que un claro ganador es *Arnold*. Una de las razones es que, como hemos visto en el capítulo 4, en la sección sobre GPUs y en el apartado “rasterization versus ray tracing”, todos los expertos parecen coincidir en que la industria se mueve hacia soluciones basados enteramente en *ray tracing*. Podríamos añadir que añadiendo luces de área y BRDFs normalizadas, y mejor aún si se utiliza con *Progressive Rendering* que permite mayor interacción. Y *Arnold* se ha hecho famoso por la calidad de su código que permite alcanzar cotas de rendimiento que parecían impensables ha ce años. Pero es interesante citar la opinión de Carlos Fajardo, el creador de *Arnold*, sobre este asunto (en <http://www.fx-guide.com/featured/the-art-of-rendering/>) comparando dos extremos: *RenderMan*, más dirigido a “pintar píxels” con todo tipo de recursos y

*Maxwell*, un sistema supuestamente “puro”. Fajardo consideraba que le interesaba situar a Arnold entre estos dos extremos. Cito: “... en un extremo tenemos a *RenderMan*, que yo describo como extraordinario pintando píxels, y en el otro extremo, algo como *Maxwell*, que es un renderer de base física muy pedante que no te deja “pintar con un pincel”, solo te deja simular la luz. Arnold estaría situado en medio, lo que tenemos es un sistema con bastante base física que te permite la simulación pero también permite mucho control artístico para cambiar cosas”. A lo que habría que añadir la gran diferencia en velocidad entre Arnold y Maxwell.

Al comentario de Fajardo cabría añadir que la fidelidad física es un objetivo que a menudo se idealiza más de la cuenta y, no pocas veces, por ignorancia de los procedimientos internos que están actuando. Pues, si tenemos en cuenta que la simulación depende de modo fundamental del modo en que se simulan las superficies materiales, resulta bastante obvio que las simplificaciones que es necesario hacer, en términos de escala progresiva, de propiedades que aparecen a diferentes niveles de aproximación, y de todas las cosas que es necesario eliminar para poder llevar a cabo una simulación creíble, estas simplificaciones son tan numerosas que la idea de “fidelidad física” se tambalea estrepitosamente. Y parece más razonable proponerse como objetivo la verosimilitud, sin importar demasiado los medios a que haya que recurrir. Otra cosa muy diferente, en lo que también destaca Arnold, es que, para un artista, resultará preferible no tener que dedicar demasiadas horas a ajustar parámetros cuya única razón de ser es técnica y que periódicamente serán substituidos por otros. Y la facilidad de interactividad es vital también desde el punto de vista de la producción profesional.

Otro criterio de ordenación más prosaico aunque no menos importante es el coste. Sin embargo la comparación no es sencilla pues algunas aplicaciones son autónomas y otras no y, más recientemente, han aparecido aplicaciones, como Clarisse o Lagoa



que proporcionan servicios parciales aunque muy interesantes. Hay aplicaciones notables, como V-Ray, que raramente se utilizan como independientes y, por lo general, requieren un programa anfitrión que puede ser de muy bajo coste como SketchUp o de un coste bastante más elevado, como 3ds Max o Maya.

Para dar alguna indicación sobre este aspecto, puedo dar alguna referencia aproximada sobre aplicaciones generales y motores dependientes.

Por lo que respecta a algunas de las aplicaciones generales, la ordenación, de menor a mayor coste sería (en 2014): *Blender* (gratuito), *Modo* (~1.200 €), *RenderMan Studio* (~1.300), *3ds Max*, *Maya* (~2.500 y 3.000).

Por lo que respecta a los motores dependientes, la ordenación de algunos de los más utilizados o más recientes (como *Octane*), de menor a mayor coste sería: *Maxwell* (75€ con SketchUp), *Octane* (~360 con 3ds Max), *V-Ray* (600€ con SketchUp, ~750€ con 3ds Max).

### RenderMan (1988, Pixar)

Los comienzos de RenderMan se remontan a la década de los 1970, en la universidad de Utah, en los 1970 y a la tesis doctoral de Edwin Catmull, uno de los fundadores de Pixar, que posteriormente pasaría a trabajar con Georges Lucas en ILM (*Industrial Light and Magic*), junto con Loren Carpenter, Robert Cook, Pat Hanrahan y otros nombres míticos en el desarrollo de todos los métodos de simulación que se utilizan en la actualidad. Allí nació REYES (siglas de “*Renders Everything You Ever Saw*”, pero también una alusión a Point Reyes, una bella región de la costa de California), el motor de *render* desarrollado inicialmente por Loren Carpenter y Robert Cook, cuando *RenderMan* era parte de *LucasFilm Computer and Graphics Research*, la división especializada en estos temas de ILM, a principios de los 1980. Se utilizó por primera vez comercialmente en la película “*Star Trek II, The Wrath of Khan*” (1982) con la secuencia del Génesis, que hizo época.

*RenderMan* nació en 1988 como una “especificación de interface” o un “lenguaje de shaders”, pocos años después de que aparecieran los primeros métodos de GI. El nombre oficial de la aplicación es *Pixar’s RenderMan* pues “RenderMan” designa propiamente al interface y el lenguaje de *shaders*. En 1986 Steve Jobs, después de abandonar Apple, compró la empresa *The Graphics Group*, subsidiaria de *LucasFilm* por 10 millones de dólares y la convirtió en Pixar que, más adelante, firmaría varios acuerdos con la compañía Walt Disney para producir películas de animación. En 2006 Walt Disney adquiriría Pixar por 7.400 millones de dólares.

Desde esas fechas, hasta la actualidad, ha sido la referencia principal, tanto desde el punto de vista teórico como práctico, para todo tipo de aplicaciones. Además de contar en su haber con una larga lista de películas pioneras en la animación digital. Entre sus colaboraciones cinematográficas que han dejado huella en la industria se pueden citar *The Abyss* (1989), *Terminator II* (1991) o *Jurassic Park* (1993), además de las series de *Toy Story* (1995, 1999 y 2010), *Monsters* (2001), *Ratatouille* (2007) o *Cars* (2011). Y muchísimas más pues Pixar ha sido probablemente la empresa de simulación más famosa de la historia de los efectos especiales y el cine de animación, además de haber ganado varios Oscar técnicos.

El motor de render REYES utilizado por Pixar ha sido también el más eficaz durante muchos años, con características especiales que luego se han ido incorporando a otras aplicaciones. Es una colección estructurada de algoritmos que utiliza prioritariamente *procedural shaders* para evitar en la medida de lo posible el uso de texturas externas, que descartaba el uso de *ray trace* para favorecer la rapidez del procesamiento y que se basaba en la subdivisión sistemática de superficies por medio de micropolígonos de un tamaño equivalente aproximadamente al de un píxel, teselando la geometría en el espacio de salida. También estaba preparado, desde los comienzos, para procesar la escena en paralelo



con múltiples ordenadores (*render farms*). También se utilizó *radiosity* como método complementario (*radiosity cache*).

Muchas de las mejoras se fueron introduciendo en paralelo a sus producciones más famosas. Así, por ejemplo, la representación del pelo por medio de primitivas especiales se añadió a raíz de “Monsters” (2001). Y, en una época en la que *ray tracing* ya se había comenzado a utilizar más extensamente a medida que aumentaba la potencia de procesamiento, el código principal se adaptó para incluir *ray tracing* con aceleración especial, a partir de 2002 aproximadamente y se se introdujo en el sistema a raíz de “Cars” (2005) y fue utilizado plenamente en películas posteriores como “Monsters University” (2013). El subsistema de *ray tracing* soporta una *cache* de teselación con multirresolución que permite captar los valores de las superficies con diferentes niveles de detalle. También incluye controles de distancia máxima que permiten optimizar la recursión del trazador. Y también incluye *múltiple importance sampling* que permite ponderar el peso relativo de las diferentes muestras. Más que substituir a los anteriores, estos métodos se sumaron a los recursos con que ya se contaba por medio de adiciones diversas, muchas de ellas basadas en procesamiento por puntos (*point-based GI*), es decir, aproximaciones a la geometría de la escena que permiten precomputar resultados y reducen el tiempo de cálculo.

Las últimas versiones posibilitan calcular la iluminación global de una escena de múltiples modos y con luces de área extensas. En general, el cálculo de la iluminación se divide en dos componentes, el cálculo de la iluminación directa, que incluye IBL y el cálculo de la iluminación indirecta que se lleva a cabo principalmente por métodos de *ray tracing* y por métodos *point-based*.

La versión 11.0 (2003) fue la primera en que se introdujo *ray tracing*, entre otras técnicas. En la versión 12.0, a partir de 2006, también se introdujo un nuevo sistema de cálculo de la iluminación, *3D baking*, un sistema de pases múltiples: *brick maps* y *point clouds*.

Estos métodos se utilizaron parcialmente en la película *Ratatouille* (2007) y extensamente en *Wall-E* (2008). Ver el resumen que se da de estos métodos en el capítulo 4 de este libro. En la versión 16.0 (2011) se incorporaban definitivamente dos sistemas de *rendering*, *ray trace* (*Raytrace Hider*) y versiones actualizadas de REYES, con el complemento de *multiresolution radiosity cache*. También incluía sombreado físico (*physical shading*). Es por tanto un sistema híbrido que es posible que se mantenga pues *ray tracing* es particularmente adecuado para cosas tales como reflexiones de entorno, reflexiones y refracciones entre objetos o sombras de color. Pero hay otros efectos que se manejan más fácil y rápidamente con métodos de rasterización. La última versión cuando esto se escribe (2014), la 18.0, sigue incorporando estos sistemas y, por añadidura, *path tracing*, *volumetric rendering*, *SSS* mejorado, *Unified Sampling* mejorado, etc.

Para más información ir a: <http://renderman.pixar.com/view/renderman>.

### Mental Ray (1989, mental images, Nvidia)

Se produjo inicialmente por *Mental Images*, una compañía con sede en Berlín, que fue fundada, en 1986, por un equipo compuesto por científicos especialistas en física y computación (Rolf Herken, Hans-Christian Hege, Robert Hödicke and Wolfgang Krüger) y economistas (Günter Ansorge, Frank Schnöckel y Hans Peter Plettner). El proyecto que desarrolló *mental ray* empezó en ese mismo año. La primera versión de *mental ray* apareció en 1989.

Puede utilizarse de modo independiente (un modo muy poco recomendable para usuarios corrientes) pero desde principios de los 1990 ha colaborado estrechamente con los principales programas de modelado y simulación 3D: con *Wavefront* y *Soft Image* (desde 1993), *3ds Max* (desde 1999 y plenamente desde 2003) o *Maya* (desde 2002). Se ha utilizado principalmente a través de estos programas aunque es una aplicación mul-





tiplataforma que también se puede integrar en otras aplicaciones. Además de *3ds Max*, *Maya* y *Soft Image*, que son sus principales anfitriones, se puede utilizar con *Houdini*, *Cinema 4D*, *Revit*, *Solid Works* o *Catia*.

A partir de 2007 pasó a formar parte de Nvidia y la compañía se renombró como *Nvidia Advanced Rendering Center* (Nvidia Arc GmbH). Su fundador Rolf Herken, sigue siendo el presidente y responsable técnico de *mental images* que en la actualidad es una empresa subsidiaria de Nvidia.

Su reputación en los últimos años es muy alta. Ganó un Oscar en 2003 (*Technical Achievement Award*) por sus contribuciones técnicas. Se ha utilizado en varias películas, entre ellas, "Hulk", "The Matrix Reloaded & Revolutions", "Star Wars Episode II: Attack of the Clones", "The Day After Tomorrow", "Poseidon"...

El motor principal se basa fundamentalmente en *ray tracing* y en la subdivisión de la escena en particiones para optimizar el cálculo. Incluye, desde sus inicios, *ray tracing* y radiosidad (según las versiones) y diversos sistemas de mejora de los tipos de luz y *shaders* especiales además de otros recursos para acelerar el tiempo de cálculo. Fue una de las primeras en incorporar con efectividad *photon mapping*. Pero el método principal desde hace varios años es *Final Gather*, solo o combinado con fotones. Puede procesar caústicas, SSS y todo tipo de efectos especiales. Las últimas versiones incorporan IBL y métodos unificados de *antialiasing*. Es un sistema abierto y programable lo que permite a usuarios avanzados incorporar sus propios *shaders*.

Técnicamente es un sistema de *ray tracing* con MIS (*multiple importance sampling*) pero es difícil decir si es *biased* o no debido a su extenso uso de *shaders* que, por otro lado, deben ser los propios de *mental ray*. Esto repercute en su eficiencia. Por un lado es extraordinariamente flexible pero por otro lado requiere cuidadosos ajustes, lo que puede repercutir en la precisión de los cálculos. Dicho de otro modo: es más complejo de lo que les gusta a muchos usuarios y requiere tiempo

para controlar adecuadamente los múltiples parámetros que intervienen. Pero, para sus usuarios avanzados, esto acaba por ser una ventaja muy importante.

Pese a todo, según la opinión de muchos expertos, *mental ray* tendrá dificultades para adaptarse a los nuevos desarrollos pues el código necesitaría reescribirse completamente (tiene una antigüedad que se remonta a 1995 o 1996) para, entre otras cosas, adaptarse a los nuevos desarrollos de la GPU, como CUDA. Sin embargo *iray* parece estar bien integrado con CUDA... Con todo, hay cosas, como el muestreo, que parece ser mejor y más rápido en V-Ray, aunque las últimas versiones de *mental ray* incorporan *Unified Sampling* que parece una mejora importante. La discusión de fondo es similar a la que se da con *RenderMan* y que he comentado más arriba: flexibilidad frente a eficacia. Es evidente que no puede haber una respuesta única a esta cuestión.

Para más información ir a: <http://www.nvidia.com/object/nvidia-mental-ray.html>.

### Cinema 4D (1993, Maxon)

Es una aplicación de modelado y simulación 3D que se desarrolló a partir de 1993 (con una versión previa con otro nombre en 1991) por la compañía alemana Maxon, en Friedrichsdorf (en el Land de Hesse). Inicialmente iba dirigida de modo exclusivo a ordenadores de tipo Amiga pero a partir de la tercera versión se adaptó también a Windows y Macintosh. Durante las primeras versiones estaba compuesta por módulos coordinados pero a partir de la versión 12 los diferentes módulos (Render, Body Paint 3d, Dynamics, Hair, Sketch&Toon, Particles, etc) se integraron en una plataforma casi única, formada por cuatro módulos con un núcleo principal.

En la actualidad hay un gran número de filiales en Francia, Japón, Singapur, Estados Unidos, Reino Unido, Canadá. Su objetivo principal es la industria de animación y se ha utilizado en un gran número de películas de todo tipo.





Utiliza principalmente dos métodos de cálculo, *Light mapping* o *Light Caching*, un método similar al utilizado por V-Ray y que es una forma aproximada de *path tracing* que se construye enviando rayos desde la cámara (al revés que con *photon mapping*) y guardando la información lumínica, registrada en cada rebote con las superficies de la escena, en un *light map*. El segundo método es una versión propia de un *physical rendering*, más avanzada y más preciso.

Permite incorporar varios motores de *rendering*, concretamente: *mental ray*, *V-Ray*, *Octane*, *Indigo*, *Pixar RenderMan*, *3D Delight*, *Maxwell*, y algunos más. Es una de las aplicaciones más abiertas en este sentido y utiliza todo tipo de métodos de cálculo en función de las necesidades o preferencias de los usuarios.

Para más información ir a: <http://www.maxon.net/>.

### Lightwave 3D (1994, Newtek)

Es una aplicación completa de simulación desarrollada por la compañía New Tek (fundada en 1985 en Kansas y especializada en *hardware* y *software* para vídeo y aplicaciones visuales). Apareció como una aplicación independiente en 1994 aunque sus inicios se remontan a 1988 cuando Allen Hastings creó un programa, Videoscapes, de *rendering* y animación, mientras su amigo Stuart Ferguson creaba otro programa complementario, Modeler. Los programas fueron adquiridos por New Tek y tras varios cambios acabaron dando lugar a Lightwave 3D.

Se ha utilizado en un gran número de películas, programas de televisión, publicidad, juegos de vídeo, etc. La lista es bastante impresionante y se puede consultar en la página oficial. Por añadidura, en 2007 se creó la primera película 3D animada por una sola persona, "Flatland the Film", por Ladd Ehlinger Jr. Ha ganado un gran número de Emmy Awards.

Se compone de dos módulos principales, *Modeler* y *Layout*. Si se utilizan los dos

programas simultáneamente hay un tercer módulo, the *Hub*, que los sincroniza. *Layout* contiene el sistema de animación y el motor de *render* que proporciona varias opciones de cálculo.

Aunque generalmente ha utilizado métodos de *scanline*, a partir de 2006 incluye un método basado enteramente en *ray trace*. Pero mantiene otros recursos tradicionales. Y el método básico es un híbrido de *ray trace* y *radiosity* cuya finalidad, como se reconoce explícitamente por la compañía, no es ser físicamente correcto sino proporcionar soluciones rápidas y verosímiles.

No incluye las formas principales de *importance sampling* aunque está prevista su incorporación en un futuro cercano. Esto limita la aplicabilidad del sistema de *ray tracing* que puede utilizar más fuerza bruta de lo que es habitual en aplicaciones de estas características. También se ha discutido, por lo que parece, la posibilidad de adaptar el sistema a *GPU rendering* si bien parece que esto no encaja con los requisitos de flexibilidad y bajo coste (relativo) del producto.

También admite la utilización de otros motores de *render* aunque no los más habituales con la posible excepción reciente de *Octane*. Por otro lado también incluye acceso gratuito a *renderfarms* para sus usuarios.

Para más información ir a: <https://www.lightwave3d.com/>

### Mantra con Houdini (1996, Side Effects Software)

Houdini es una aplicación de animación 3D desarrollada por la compañía de Toronto, Side Effects Software. Su primera versión apareció en 1996 y estaba basada en un producto anterior (Prisms) desarrollado igualmente por Kim Davidson y Greg Hermanonvi, los fundadores de la compañía, y que también estaba basado en métodos procedurales, una de las características de Mantra, el motor de *render* que utiliza Houdini.

Mantra es un motor de *render* muy similar a *RenderMan*, cuyos primeros pasos se re-



montan a 1988, en productos previos desarrollados por Mark Elendt que luego pasó a integrarse en la compañía. Utiliza una combinación de métodos, principalmente *micropolygon rendering*, un método híbrido de *scan-line* y *ray trace* y módulos especiales para cálculo volumétrico (humo, nubes, etc). El método de micropolígonos está basado en los algoritmos de REYES que, como hemos visto, era y sigue siendo uno de los recursos básicos de *RenderMan*, un algoritmo del tipo “divide and conquer”, una estrategia clásica para abordar un problema por medio de subdivisiones sistemáticas hasta llegar a un nivel en que el problema complejo se ha convertido en un problema simple. El método de procesamiento actual está basado en *micro-voxel rendering*, lo que ha permitido procesar de un modo más eficaz el tipo de efectos por los que *Mantra* ha adquirido su prestigio, efectos tales como fuego, humo, niebla, gases, etc. La principal reputación de *Houdini/Mantra* deriva de sus métodos de *volumetric rendering*. También soporta SSS por medio de nubes de puntos e *irradiance cache*. Han ganado varios Oscars técnicos por sus trabajos en este área en varias películas.

Para más información ir a: <http://www.sidefx.com/>.

### V-Ray (1997, Chaos Group)

La aplicación *V-Ray* se desarrolló a partir de 1997 por la empresa Chaos Group, en Sofía, Bulgaria. Los desarrolladores principales fueron Vladimir Koylazov y Peter Mitev. Desde esa fecha, en la que el número de miembros se limitaba a dos, ha ido creciendo hasta agrupar a más de 100 personas que trabajan en oficinas dispersas en Europa, Estados Unidos, Japón y Corea. A lo largo de este periodo se ha convertido en uno de los motores de *render* más populares de todo el mundo y ha sido utilizado en una gran cantidad de películas de todo tipo en Europa y Estados Unidos y ha ganado varios premios internacionales. Es muy apreciado por su combinación de calidad y rapidez.

Aunque en las últimas versiones se puede usar de modo independiente se utiliza por lo general a través de una aplicación anfitriona. Las más corrientes son *3ds Max*, *Maya*, *Soft Image*, *Rhino*, *SketchUp* y *Blender*, entre otras. En las últimas versiones también se ofrecen dos productos autónomos, uno, *V-Ray*, basado en la CPU y otro, *V-Ray RT GPU*, basado en la GPU.

Es un método flexible que puede procesar la escena con métodos *biased* o *unbiased*, según lo que se necesite en cada caso. El procedimiento básico se basa en *ray tracing* pero una de las características de *V-Ray*, que se analizarán con detalle en las siguientes secciones, es que permite combinar de diversos modos cuatro métodos principales de cálculo: *irradiance*, *photons*, *light cache* (un tipo de *path tracing* con *progressive rendering*) y fuerza bruta (QMC, *Quasi Monte Carlo* o DMC, *Deterministic Monte Carlo*).

Incluye varios recursos avanzados, mapas de desplazamiento de alta calidad, muy eficientes, SSS *shaders* excelentes, *shaders* especiales muy eficaces para *hair and fur* (pelo y piel), *shaders* especiales para reflejos metalizados (coches, etc) que unifican los *shaders* con múltiples capas en uno solo, etc. Todos estos recursos se encuentran igualmente en otros programas pero *V-Ray* los gestiona con particular eficacia. Incluye también IBL con una herramienta especial, *Dome Light* que proporciona un procesamiento de IBL mejor que el de muchos otros motores.

También incluye algunos recursos útiles para escenas de grandes dimensiones, *proxy meshes*, también denominados *tiled meshes*, que se utilizan para preprocesar la geometría, creando un archivo de tipo “.vrmesh” en disco que puede cargarse parcialmente al hacer un *render* sin necesidad de guardarlo todo en RAM. Otro tanto puede hacerse con grandes texturas, preprocesándolas y guardándolas como *tiled textures*. Se pueden aplicar a través de OpenEXR que soporta este recurso de modo nativo. Esto permite utilizar versiones de baja resolución para objetos lejanos.



Como ocurre con todos los sistemas que utilizan *ray tracing* de uno u otro modo, *Importance Sampling* es una parte fundamental de la aplicación para asegurar que no se desperdician cientos de miles de rayos enviados en direcciones sin interés. Esto requiere precálculos habituales pero que están bien integrados en *V-Ray*. Se utiliza MIS (*Multiple Importance Sampling*) combinando diferentes técnicas de muestreo, principalmente para luces de área y *dome lights*. También se utiliza un método muy eficaz de muestreo adaptativo, *Adaptive DMC*, que toma un número variable de muestras por píxel a partir de las diferencias de intensidad entre píxeles vecinos, un método particularmente efectivo en escenas con muchos detalles finos. Es un método que funciona prácticamente de modo automático y no requiere grandes ajustes por parte del usuario.

Para más información ir a: <http://www.chaos-group.com/en/2/index.html>.

### 3Delight (2000, DNA Research)

3Delight se desarrolló por la compañía DNA Research, subsidiaria de Taarna Studios, una empresa internacional con oficinas centrales en Singapur, y su primera versión apareció en 2000. Fue la primera aplicación que se adaptaba estrictamente a *RenderMan*, combinando los algoritmos de REYES con *ray tracing* en función de los intereses del usuario. Las primeras versiones fueron gratuitas. A partir de 2005, cuando la aplicación se había extendido suficientemente, las licencias, para dos estaciones, pasaron a costar 1.000 dólares (y 1.500 para cuatro).

Se ha considerado como una especie de “versión pobre” de *RenderMan*. De hecho incorpora todas las capacidades principales de *RenderMan* e incluye luces de área, *volume shading*, *shading* programable, *path tracing*, SSS, etc. Las últimas versiones incluyen MIS y nuevos materiales y *path tracing* bien optimizado y muy rápido. Está basado en módulos.

Se presenta en tres modos especiales, como aplicación independiente (sobre Win-

dows, Linux o Mac) y como aplicación ligada a *Maya* y *Soft Image*.

Se ha utilizado para efectos especiales en un gran número de películas, como “Happy Feet Two” (2001), junto con *Maya* y *Mantra* (*Houdini*). Su uso en producción cinematográfica está más extendido fuera de Estados Unidos.

Para más información ir a: <http://www.3delight.com/en/index.php>.

### FinalRender (~2001, Cebas)

*FinalRender* está producido por Cebas Computer (una compañía de Canadá, en Victoria, British Columbia) que desarrolla plugins para 3D Studio y otros programas desde 1989. En la actualidad incorpora a desarrolladores y personal diverso en Australia, Alemania y Rusia, además de Canadá. Se ha utilizado parcialmente en varias películas conocidas para generar efectos especiales (“Harry Potter”, “Spiderman 3”, “2012”, “Alice in Wonderland”) y en varios juegos de vídeo.

La primera versión, *Stage-0*, creada para la versión 4.0 de *3ds Max*, hacia 2001 (no he podido encontrar la fecha exacta) solo funcionaba con *3ds Max*. A partir de la tercera versión, *Stage-2*, que apareció en 2005, también se incluía soporte para *Maya* y *Cinema 4D*. No funciona de modo independiente sino como plugin para un programa anfitrión que, en principio es *3ds Max*, *Maya* y *Cinema 4D*.

Incorpora radiosidad y *ray tracing* además de otros complementos como efectos cáusticos, reflexiones y refracciones mejoradas y, entre otras cosas, un tipo de luz cilíndrica que es particularmente adecuado para representar tubos fluorescentes. Es claramente más rápido que el motor predeterminado de *3ds Max*. En las últimas versiones permite utilizar varios métodos de iluminación Global, principalmente: *Irradiance Caching*, métodos *Unbiased*, o *Light Cache*.

En los últimos años se ha modificado radicalmente, con nuevo código y nuevas estrategias técnicas, y fue una de las primeras aplicaciones que aplicó métodos de cálculo



de iluminación global a una película de larga duración, “2012”.

La última versión, que está prevista que aparezca en 2014, incluirá soporte para la GPU, con un uso extenso de CUDA que lo convierte en un producto prácticamente nuevo. La finalidad principal es aumentar de modo espectacular la rapidez de cálculo. De hecho es una estructura híbrida que utiliza extensamente la GPU pero sigue relegando ciertos aspectos del cálculo a la CPU. Incluye un número ilimitado de *Render Elements* en diferentes capas, *Volumetric Shaders*, Materiales compuestos complejos y un uso amplio de formatos OpenEXR. También permite combinar el procesamiento con otros materiales, concretamente con los materiales arquitectónicos de *mental ray* que ofrecen numerosas ventajas.

La versión *Final Render 4 GPU* ofrece tres métodos de *rendering*: a) solo GPU, b) solo CPU (como en la versión previa, la 3.5), c) CPU+GPU, un método híbrido que procesa el modelo inicialmente con la CPU pero tiene la capacidad de derivar ciertas líneas de cálculo a la GPU, incluso si el proceso está basado en un plugin que solo está preparado para ser procesado con la CPU. Esto le permite, entre otras cosas, sacar partido del gran número de plugins que hay disponibles para *3ds Max*, que sigue siendo el programa anfitrión principal para *Final Render*. Pese a esto hay algunas limitaciones pues Cebas también produce *Thinking Particles* uno de los mejores sistemas para simular fuego pero que no funciona adecuadamente con *3ds Max*.

Para más información ir a: [www.cebass.com](http://www.cebass.com)

### Modo (2004, The Foundry)

The *Foundry Visionmonger* es una compañía de desarrollo de aplicaciones para composición y efectos. Su sede principal está en Londres y tiene delegaciones en Los Angeles y Redwood City, California. Se fundó en 1996 y, en 2007 fue adquirida por *Digital Domain* lo que permitió incorporar a *Nuke*, probablemente la plataforma principal para composición de

efectos especiales de los últimos años. Posteriormente pasó a otros propietarios (Carlyle Group, en 2011) y, en 2012 se fusionó con *Luxology*, otra compañía de desarrollo de aplicaciones para simulación 3D y animación cuyo principal producto era *Modo*. Otros productos de interés relacionados con *Modo* son *MARI*, una aplicación de pintura digital que apareció en 2010 y que se utilizó extensamente en la película “Avatar”, y *Katana*, una herramienta de simulación de iluminación desarrollada inicialmente por Sony y que apareció en 2011.

*Modo* se creó por el mismo equipo principal que creó *Lightwave 3D*, desarrollado originalmente para la plataforma Amiga. En 2001, a raíz de la propuesta de reescritura completa del código y renovación completa de los recursos tecnológicos de *Lightwave 3D*, Brad Peeble, uno de los miembros principales, junto con Allen Hastings y Stuart Ferguson, los desarrolladores principales y otros miembros del equipo, dejaron *Lightwave* para fundar *Luxology*. Después de tres años de trabajo presentaron *Modo* en Siggraph 2004 y el mismo año aparecía la primera versión de la aplicación. En 2006, la compañía *Digital Domain* integró a *Modo* en su línea de producción y posteriormente pasó a ser utilizada por otras compañías líderes del mercado como Pixar, ILM, Zoic Studio y otras. Durante los siguientes años aparecieron nuevas versiones, se ganaron premios y su posición se consolidó. La versión más reciente es la 701 (2013).

Una de las principales diferencias con otros programas, como *3ds Max* o *Maya*, que incluyen una gran colección de recursos para diferentes finalidades, es que *Modo* utiliza una colección pequeña de recursos básicos que pueden combinarse para crear nuevas herramientas, por medio de acciones personalizadas coordinadas por la *Tool Pipe* que permite escoger *Actions* (que definen procesos) y *Falloffs* (que ajustan la mayor o menor influencia de una acción). Otra herramienta notable es *3D Painting* que permite pintar directamente sobre modelos 3D o pintar instancias de mallas previas sobre la superficie de otro objeto con una colección variada de herramientas.



El motor de *render* principal está basado en *ray tracing* físicamente correcto (*unbiased*), es *multithreaded* y puede escalarse de un modo casi lineal para adaptarse a un número diferente de núcleos (es decir que un ordenador con 8 núcleos procesaría una imagen 8 veces más rápido, aproximadamente, que otro con un solo núcleo). El motor admite hasta 32 núcleos y también puede computarse en red. También incluye un módulo de previsualización progresiva (*progressive render*) que puede continuar hasta alcanzar la calidad máxima deseada. Es un motor de *render* híbrido que, con la configuración adecuada puede funcionar como un método *unbiased*, físicamente correcto. Pero también puede incorporar *irradiance caching* para acelerar los cálculos. Sus creadores buscaban la versatilidad de *RenderMan* pero incorporando las nuevas técnicas que se estaban desarrollando por esas fechas. Incluye cáusticas, dispersión, rendering estereoscópico, reflexiones Fresnel y refracciones dispersas, SSS, *volumetric rendering*, etc... Los métodos de IS (*Importance Sampling*) son quizá algo limitados en comparación con otras aplicaciones. También incluye *toolkits* que puede extender sus capacidad para hacer cosas tales como NPR (*Non Photorealistic Rendering*). A veces se utiliza conjuntamente con un motor externo, en la mayoría de los casos, *Maxwell* o *V-Ray*. Por otro lado, la compañía *The Foundry* está experimentando con *Modo* y *Nuke* conectados, lo que amplía notablemente las posibilidades.

Para más información ir a: <http://www.thefoundry.co.uk/products/modo/>.

### Maxwell (2006, Next Limit)

La primera versión de *Maxwell*, apareció en 2006, tras un par de años de desarrollo experimental, producido por la compañía Next Limit, una compañía fundada en Madrid en 1998 por Víctor González e Ignacio Vargas. Sus creadores buscaron, desde el comienzo, que su sistema de cálculo fuera "físicamente correcto". Para ello se basaron en una forma específica de *path tracing bidireccional*, MLT

(*Metropolitan Light Transport*) que ya hemos visto y volveremos a ver más adelante. En 2009 apareció la versión 2.0 y en 2013 la 3.0. El director actual de *Maxwell Render Technology* es Juan Cañada.

Puede funcionar como aplicación independiente pero se utiliza principalmente a través de aplicaciones anfitrionas, prácticamente todas las dignas de mención: *3ds Max*, *ArchicAD*, *Artlantis*, *Autodesk Revit*, *Bonzai 3D*, *Cinema 4D*, *Form-Z*, *Houdini*, *Lightwave*, *Maya*, *Microstation*, *Modo*, *Rhino3D*, *SketchUp*, *Softimage XSI*, *Solid Works*. De hecho, un objetivo que se buscó desde el comienzo es que fuera compatible con cualquier plataforma y, en consecuencia, independiente de cualquier adaptación a una GPU concreta. Esta ventaja implica una importante desventaja pues el principal inconveniente de *Maxwell* es que es muy lento ya que se basa enteramente en la CPU y no saca prácticamente ningún partido de los avances de las GPU actuales.

También incluye un módulo denominado FIRE, un módulo de previsualización que se basa en *progressive rendering* lo que facilita el trabajo y permite interrumpir el cálculo si los resultados iniciales no son correctos.

Su mayor cualidad es la alta precisión de cálculo lo que da lugar a imágenes que, en ocasiones se utilizan como referencia para juzgar la calidad de otros motores de *render*. Sus objetivos iniciales no estaban dirigidos a aplicaciones comerciales sino que se buscaba obtener imágenes realistas precisas, sin consideración por el tiempo empleado. Incluye *volumetric rendering*, *procedural maps*, *deep compositing*, cáusticas de precisión, etc. Una de sus ventajas es que no requiere ajustar múltiples parámetros.

Aunque los creadores reivindican la exactitud y la fidelidad a las propiedades de los objetos físicos, también hay que repetir, como lo he hecho en la introducción a esta sección, que esta es una pretensión discutible pues todos los materiales están basados, como los de cualquier otro programa, en *shaders* que simulan propiedades diversas de modo independiente por métodos inevitablemente artifi-





ciosos, o que utilizan mapas de bits externos para simular texturas. Y esto no es precisamente lo que ocurre en el mundo real. Otro tanto cabe decir de la simulación de la iluminación pues, en la práctica, hay que recurrir a diversos artificios para simular distribuciones reales.

Con todo, una vez que el objetivo inicial, el intentar conseguir resultados de alta calidad sin preocuparse por el tiempo de cálculo, se ha conseguido y la reputación de la aplicación se ha consolidado, parece que la estrategia se está dirigiendo a conseguir mayor velocidad y efectividad. Por ello, en las últimas versiones se han introducido métodos de MIS (*Multiple Importance Sampling*) y *multicore threading* para acelerar el cálculo así como Open CL para poder sacar partido tanto de las GPU de Nvidia como las de AMD/ATI.

Para más información ir a: <http://www.maxwellrender.com/>.

### Arnold (2009, Solid Angle)

Los inicios de Arnold se remontan a 1997, año en el que Marcos Fajardo, nacido en España pero que vivía en Estados Unidos, con 24 años decidió desarrollar su propio motor de *render* (el nombre parece que deriva de las bromas de un amigo mientras veían una película de Arnold Schwarzenegger, aunque inicialmente se denominó “Render API”, un nombre bastante aburrido, como le hicieron ver estos amigos). Fajardo había estado trabajando en este motor desde 1994. En 2001, después de combinar *Arnold* con un plugin para *3ds Max* con resultados más que notables, estuvo trabajando con Paul Debevec en un proyecto para el Partenón que se representó con *Arnold*. Entre 2004 y 2009 Fajardo trabajó en Sony Pictures pero mantuvo la propiedad del *software* y en 2009 fundó *Solid Angles*, en Madrid, para desarrollar *Arnold* de modo independiente. En 2012 la compañía contaba con 16 colaboradores y ha seguido creciendo aceleradamente desde esa fecha. El nombre de la compañía deriva, por lo que parece, de la definición de *radiance* que apa-

rece en la *rendering equation* de Kajiya como “amount of light that is emitted from a particular area, and falls within a given *solid angle* in a specified direction”.

A diferencia de otras aplicaciones, *Arnold* creció desde arriba, dirigiéndose de entrada a las grandes firmas, razón por la que es relativamente poco conocido en el mundo de los ordenadores corrientes. Una de las primeras compañías en utilizar *Arnold* ha sido *Entropy Studio* a la que han seguido muchas más. Del éxito y el crecimiento espectacular de este *software* puede dar una idea el que, de las 10 nominaciones técnicas a los Oscar de efectos especiales de 2014, había 7 desarrolladas integra o parcialmente con *Arnold*. Su reputación técnica entre los expertos es también muy alta.

Esta reputación es debida a un uso extraordinariamente eficiente de las estructuras de precomputación (*grids* uniformes que se despliegan en jerarquías de otras *grids*, *voxels* cúbicos de número bien adaptado a cada caso, etc). *Arnold* es, a lo largo de sus más de 200.000 líneas de código en C++, un ejemplo perfecto de *Monte Carlo ray tracing*, *unbiased* y unidireccional. A diferencia de otros programas, entre ellos *RenderMan*, utiliza *ray tracing/path tracing*, tanto para la iluminación directa como para la indirecta. Pero a diferencia de todos los demás, no es lento. Entre otras cosas porque no usa pasos previos de precomputación, como en *irradiance cache* o métodos similares. Y al usar *progressive rendering* se facilita la interacción y la cancelación de un resultado que pronto se ve que no interesa. Tampoco utiliza trucos ni docenas de parámetros para ajustar o corregir errores. Una ventaja general de *path tracing* es que no requiere utilizar *shaders* complejos.

Puede decirse que *Arnold* tiene básicamente un sólo parámetro: el número de muestras. Resulta por tanto muy sencillo de usar y no consume memoria, que se reserva para almacenar gigantescas estructuras de polígonos (de 100 a 200 millones en algunas escenas de “Alice in Wonderland”). Por otro





lado, estaba basado enteramente en la CPU, sin recurrir a la GPU (como Maxwell). Esto es una limitación importante pero facilita su uso con *render* distribuido (en *render farms* en *render clouds*, vía Internet) y no necesita adaptar el código a diferentes plataformas.

Incluye SSS (con *ray tracing* en lugar de *point clouds*), MIS (*Multiple Importance Sampling*), *Multithreading* (que permite aumentar la velocidad de *ray tracing*, reducir el ruido y superar las limitaciones de memoria), *volumetric rendering* y *volumetric lighting*...

La lista de películas recientes que han utilizado Arnold es muy larga y se puede consultar en la página oficial (entre otras "Monster House", "Alice in Wonderland", "Thor", "Captain America", "X-Men: First Class", "The Avengers", "Red Tails", "Space Pirate Captain Harlock", "Elysium", "Pacific Rim", "Gravity"...). Fue el programa que se utilizó para aplicar *path tracing*, un método de fuerza bruta impensable hasta entonces como recurso para películas animadas, en "Monster House" (Sony Pictures) en 2006.

No es un producto disponible, al menos por el momento, para ordenadores corrientes pues los clientes de *Arnold* son grandes compañías y no parece haber un interés a corto plazo por comercializarlo de un modo más extenso. Pero parece evidente que marca una línea de desarrollo que tendrá una influencia considerable en las nuevas aplicaciones.

Para más información ir a: <http://www.solidangle.com/>.

### Octane (2011, Otoy).

Es un producto muy reciente basado enteramente en la GPU, tanto con base local como en la nube. De hecho, ha sido el primer *render* físicamente "correcto" (*unbiased*) que funciona de modo exclusivo sobre la GPU. Funciona, también de modo exclusivo, con la tecnología CUDA de Nvidia. Su gran ventaja con respecto a otros productos es precisamente que se ha diseñado teniendo en cuenta esta nueva tecnología que ha aparecido en los últimos años.

Se produjo inicialmente (en versión beta) por la compañía *Refractive Software*, de Nueva Zelanda, por su fundador, Terrence Vergauwen y en 2012 fue adquirido por OTOY que lo comercializó como aplicación independiente y como plugin para *3ds Max*, *Maya* y otras aplicaciones. La última versión incluía plugins para *ArchiCAD*, *Blender*, *Daz Studio*, *Lightwave*, *Poser*, *Rhino*, *3ds Max*, *AutoCad*, *Cinema4D*, *Inventor*, *Maya*, *Revit*, *Softimage*, *SketchUp* (en desarrollo), *Modo* (en desarrollo), *Carrara* (en desarrollo).

La compañía sigue trabajando con el equipo inicial de Nueva Zelanda y, por otro lado, mantiene enlaces importantes con empresas que comercializan la computación en la nube (*cloud computing*) y también es propietaria de Lightstage. También están relacionados con Autodesk que proporciona una vía adicional para el desarrollo de plugins para varias aplicaciones. OTOY tiene (en 2014) cuatro oficinas, dos en Los Ángeles (incluyendo Lightstage) y otras dos en Nueva Zelanda.

La compañía también es propietaria de *Brigade*, otra aplicación que, cuando esto se escribe, está aún en fase experimental y está basada en un *renderer* con *path tracing* pero muy rápido. Aunque *Brigade* y *Octane* son productos diferentes tienen bastantes cosas en común incluyendo, por lo que parece, varios algoritmos y recursos.

Es muy probable que permita incorporar a ordenadores personales la tecnología que hasta ahora solo estaba disponible para las grandes empresas. *Octane* funciona sobre PC aunque hay un interés prioritario estratégico para facilitar su aplicación en *GPU cloud render farms*, con GPU especiales desarrolladas por Nvidia. La idea es que los usuarios puedan elegir en cualquier elemento entre hacer un *render* en su ordenador local o enviarlo a la nube. El escalado será lineal, es decir que si se quiere utilizar 100 o 1000 GPUs el tiempo se reducirá proporcionalmente. Y es de suponer que el coste por hora también aumentará proporcionalmente, aunque la compañía ya ha anunciado que será inferior a otras ofertas similares ya disponibles (Ama-



zon ofrecía computación a distancia a unos 2 \$/hora en 2013).

Utiliza materiales y sistemas de iluminación físicamente “correctos” (*unbiased*), *shaders* SSS, cámaras realistas, etc.

Conviene seguir las recomendaciones de OTOY para usar una GPU que saque pleno rendimiento del programa. En 2013 las recomendaciones eran de tarjetas de gama media/ alta como las Geforce GTX570 o GTX580. Las últimas tarjetas de Nvidia, basadas en arquitectura Kepler no funcionaban tan bien como las basadas en Fermi aunque se estaba trabajando para adaptar el programa a los nuevos sistemas.

Para más información ir a: <http://render.otoy.com/>.

### Clarisse (2012, iFX Isotropic)

Fundada por Sam Assadian y Sebastien Guichou hacia 1996, Isotropix es una compañía francesa con sede en Montpellier.

Puede considerarse como una combinación de un *3D renderer*, un *2D compositor* y un módulo de animación. Se utiliza no como un recurso final sino como un recurso de trabajo que se inserta en un determinado punto del flujo de producción, para facilitar el trabajo de los diseñadores creando imágenes de diferentes elementos (diferentes capas o *render elements*). Es una combinación de aplicaciones de simulación y composición. Se diseñó para centrarse en la imagen como requisito principal, de tal modo que se pudiera trabajar directamente sobre ella.

El núcleo principal incluye un motor de *render* que puede procesar el resultado en milisegundos. Por otro lado no tiene recursos de modelado, está diseñado para importar modelos y procesarlos con rapidez.

Está basado en *path tracing* aunque está previsto que en versiones posteriores incorpore también *irradiance caching*, de nuevo, por razones de rapidez. No incluye caústicas ni otros efectos más o menos corrientes en otras aplicaciones. Es *multithreading*.

Para más información ir a: <http://www.isotropix.com/>.

### Lagoa MultiOptics (2013, Lagoa)

Se desarrolló por un equipo de científicos, ingenieros y diseñadores con oficinas en Montreal, Boston y Bonn, fundado por Arno Zinke, Thiago Costa y Dov Amihold. La primera versión comercial se lanzó en 2013.

En este caso se trata de un *renderer* que funciona exclusivamente en la nube, a través de la web, con un interface sencillo que facilita el envío de archivos. Utiliza varios recursos que se basan en materiales que la compañía denomina *multi-optics*. Esto quiere decir que hay un recurso específico para, por ejemplo, simular pelo y otro para simular piel con SSS.

El procedimiento básico está basado en el motor de *MultiOptics*, un sistema de *render* de fundamentos físicamente correctos (*unbiased*), basado en *ray tracing*. No requiere precómputo y puede adaptarse a todo tipo de complejidades del modelo. Con todo, los desarrolladores, rechazan la división habitual entre *biased* y *unbiased* y consideran que la palabra clave es “consistencia”. Se utiliza *importance sampling*, tanto para materiales como para fuentes de luz e IBL. Hay un trabajo importante en investigación sobre *shaders* de materiales y además de la BRDF clásica también se utiliza BCSDf (*Bidirectional Curve Scattering Distribution Function*) y BTF (*Bidirectional Texture Function*). Incluye también una biblioteca de materiales de todo tipo, desde plásticos simples a SSS, que está basada en mediciones físicas de materiales reales equivalentes. Las cámaras son de tipo físico.

El procedimiento consiste en que se envía el modelo en un formato standard, como OBJ o 3DS, con las texturas en cualquiera de los formatos corrientes (jpg, png, hdr, exr...). Se puede usar cualquier tipo de luz o geometría o entorno (IBLI) como fuente de iluminación. Utilizan un sistema de compresión de archivos propio y todos los archivos que se envían se convierten a este formato. Dado que el *render* se lleva a cabo en un equipo remoto se puede seguir trabajando mientras se procesa el resultado.



Ofrecen (2014) una versión gratuita con 2 horas de computación y 1 Gb de almacenamiento al mes. La versión de pago cuesta 50 \$ al mes y no tiene límite. Hay versiones más caras con mayor poder de computación.

Para más información ir a: <http://home.lagoa.com/>.

## Escenario utilizado en los análisis de este capítulo

Para facilitar la comparación, la gran mayoría de los ejemplos de este capítulo se desarrollan con un mismo ejemplo que se resume en este apartado.

**Geometría.** La escena consiste en un espacio de 3x3x3 metros en el que hay situados dos prismas cuyo tamaño y disposición están inspirados en la escultura minimalista de Robert Morris “Danza” (1961). El espacio es similar a la *Utah Box*, otro recinto de referencia utilizado en varias pruebas de simulación y tanto el espacio y la disposición de los prismas también son similares a los utilizados por Dutré, Bala y Bekaert en otra obra de referencia notable de 2006 sobre iluminación avanzada (ver la bibliografía). También hay un objeto que actúa como luminaria, un prisma de 120x120x3 cm situado en el centro del recinto a 295 cm de altura.

**Materiales.** Todos los objetos tienen asignado un material simple, no reflectante, de color gris medio (rgb 120, 120, 120 en un rango de 0 a 255 o 0.588, 0.588, 0.588 en un rango de 0.0 a 1.0), excepto el prisma que simula una luminaria que tiene asignado un material autoiluminante.

**Luces.** Hay una única luz fotométrica de área, de 120x120 cm, situada unos 2 cm por debajo del prisma que simula la luminaria.

**Cámaras.** Hay una única cámara situada a 150 cm de altura, apuntando al centro del recinto, desde el exterior. Para que el interior sea visible hay un plano de recorte a ras del muro frontal. De este modo en las simulaciones de cada sistema se tendrán en cuenta los rebotes sobre el muro que no vemos, situado frente a la cámara.

Tener en cuenta que, cuando se utilice *mental ray* con 3ds Max hay que mantener activado el algoritmo *scanline* del motor predeterminado de 3ds Max para que se procesen los planos de recorte: si la imagen apareciese negra, lo que indica que no se procesa el interior, ir a *Render Setup/Renderer/Rendering Algorithms* y marcar esta opción que, por lo general, se desactiva cuando se utiliza *mental ray* pues no es necesaria y al desactivarse permite que *mental ray* haga mejor uso dinámico de la memoria.

En algunos casos puede que haya que hacer algún ajuste de esta configuración general. Cuando esto sea así se indicará al comienzo del subcapítulo.

La descripción anterior de la escena es válida tanto para 3ds Max como para *SketchUp* con *V-Ray* o *Maxwell*. Los ejemplos dados para *V-Ray* se han procesado desde *SketchUp* con *V-Ray*.

No entro en el detalle de como preparar la escena en cualquiera de estos programas pues doy por supuesto que el lector ya sabe como hacerlo. En cualquier caso, no requiere más que un conocimiento muy básico de como utilizar estos programas. Sin embargo en el caso de *V-Ray* con *Sketchup* conviene dar alguna indicación adicional pues hay alguna diferencia que puede afectar algo a los cálculos.

A diferencia de como se ha organizado en 3ds Max, que contenía una cámara con planos de recorte para ver el interior desde el exterior, he alargado el espacio (hasta unos 7.5 cm de largo en total) pues *V-Ray* con *SketchUp* no reconoce, por lo que parece, los planos de sección de *SketchUp* y tampoco tiene medios para crearlos. Esto podría crear alguna diferencia en la iluminación global pero el análisis de los ejemplos no muestra diferencias demasiado significativas.

Luego he creado una luz rectangular de las mismas dimensiones que la de 3ds Max (120x120 cm) (he tenido que girarla 180° pues al crearla apunta hacia arriba y también he tenido que crear líneas y guías de apoyo pues no me deja o no sé, darle dimensiones



adecuadas al crearla en la antigua versión que he utilizado) y la he llevado a la posición coincidente con la luminaria original. Después he borrado esta luminaria, pues las luces de área de V-Ray se ven bien en el render. La luz de área V-Ray está ligeramente más elevada que la luz fotométrica del archivo original (en 2.90 en lugar de 2.85) para compensar por la eliminación de la falsa luminaria.

La intensidad de esta luz se ha ajustado con el multiplicador escalar (escoger *default* en la lista colgante de tipos de unidades) ajustado a un valor de 100.0 (el valor predeterminado es 50.0). Aunque V-Ray permite escoger entre cuatro tipo de unidades, su correspondencia con las reales no funciona muy bien y no se corresponde con las de otros programas. Por otro lado, en unos casos es independiente del tamaño (*lumens power* y *radiant power*) y en otros no (*luminance* y *radiance*) lo que complica aún más el ajuste.

En *Options/Camera*, he mantenido una cámara de tipo estándar pero con la opción *override FOV* activada y un valor para FOV de 45°. La configuración de la cámara física (de tipo “still”), que afectan a la exposición y por tanto a la mayor o menor luminosidad del resultado, son los siguientes (sólo he cambiado el primero, la velocidad de obturación, con respecto a los predeterminados): *Shutter speed* 100.0. *F number* 8.0. *Film Speed* 100.0. El resto de los valores se han dejado como estaban.

En *Options/Output* he dado como valor de salida 512x512 para pruebas y 1024x1024 para los resultados finales.

En *Options/Image Sampler* se controlan los valores generales de *antialiasing*. He seleccionado *Adaptive Subdivision* que da mejores resultados y mejora el *antialiasing* en líneas verticales y horizontales, con valores de *Min Rate* -1, *Max Rate* 2 y *Color threshold* 0.15. También he marcado la opción *Object Outline* que suaviza los bordes escalonados de líneas inclinadas.

Estos son ajustes generales. Los específicos se verán más adelante, en cada caso.

## 6.2 Radiosidad

### Radiosidad con 3ds Max

Los principios generales ya se han descrito en el capítulo 3. En esta sección resumo los pasos que hay que dar, en 3ds Max, para obtener el resultado que se muestra en la última figura de este capítulo.

Al activar un cálculo por radiosidad lo habitual es ajustar los parámetros teniendo en cuenta tres estadios básicos:

1) Calidad inicial. La configuración inicial depende de un precálculo estadístico que establece un número mínimo de pasos para obtener una primera aproximación al resultado;

2) Refinamiento de iteraciones y ajuste de la densidad de la malla. Un resultado corriente tras el primer estadio es una escena en la que aparecen zonas oscuras porque ningún rayo las ha tenido en cuenta. En este segundo estadio es posible hacer que el sistema vuelva a repetir los cálculos teniendo en cuenta la contribución de nuevas zonas, mediante la especificación de más rebotes de la luz o forzando una subdivisión más fina de la malla y computando su contribución al resultado provisional. Esto puede hacerse para toda la escena o para objetos seleccionados;

3) Recolección (*regathering*). A pesar de proceso de refinamiento anterior aparecen errores (bordes escalonados o discontinuidades características) debido principalmente a las limitaciones geométricas del modelo. Para eliminar, en la medida de lo posible estos errores, se puede activar un tercer tipo de corrección denominado *pixel regathering* que ajusta los valores para cada píxel en el momento de la representación. Esto lleva bastante tiempo por lo que la opción más recomendable es proceder ordenadamente y no activar este estadio hasta que los dos anteriores no hayan proporcionado resultados mínimamente satisfactorios. Puede ser innecesario, de todos modos, según los casos, utilizar los refinamientos propios del segundo estadio si se piensa activar esta opción. Debe tenerse en cuenta que el tiempo de cálculo crece de



modo exponencial por lo que el último 5% de cálculo, en el que se van ajustando matices de iluminación puede tardar tanto como el 95% inicial que puede dar ya una aproximación bastante correcta al resultado ideal.

### Preparación. Escenario utilizado. Sumario del método

El proceso básico, en Max, para una escena sencilla, como la que estamos utilizando en todos los casos, sería el siguiente.

1. Organizar las luces y la escena adecuadamente. Esto ya se ha descrito en la introducción. Pero es especialmente importante con radiosity asegurarse de que las unidades son correctas pues se trabaja con unidades fotométricas reales. En nuestro ejemplo la intensidad de la luz se ha establecido en 1500 cd.
2. Si está activado *mental ray*, como es corriente en las últimas versiones de este programa, no estará disponible Radiosity. Hay que escoger *Default Scanline Renderer* como motor de Render. Para ello ir a *Render Setup/Common/Assign Renderer* y descolgar la lista de motores disponibles. Tras hacer esto las pestañas de

*Render Setup* cambiarán y aparecerá una nueva, *Advanced Lighting*.

3. Cambiar la configuración de sombras de la luz. Con *mental ray*, como veremos más adelante, la configuración más adecuada es sombras de *ray trace* con luces extensas. Sin embargo con el motor predeterminado de Max esto no funciona por lo que hay que hacer un pequeño cambio: seleccionar la luz, activar sombras de área y hacer que el tamaño del área asignada a la sombra sea del mismo tamaño que la luz (120x120 cm). El resultado obtenido, será similar.
4. En el cuadro de diálogo de *Advanced Lighting*, escoger *Radiosity* y activarlo. Al hacer esto se desplegarán cinco persianas de las que, para lo que sigue, nos interesan principalmente la primera (*Radiosity Processing Parameters*), la segunda (*Radiosity Meshing Parameters*) y la cuarta (*Rendering Parameters*). La tercera (*Lighting Parameters* sirve para asignar intensidades de iluminación como si se pintase (ver la Ayuda de Max si interesa probar esta opción). Y la quinta (*Statistics*) solo sirve para dar información (que merecerá la pena comprobar). También se puede entrar en este cuadro a través de los menús contextuales (*Quad menus*), haciendo *BDR+Alt+Ctrl/ Radiosity*.
5. En *Render Setup/ Advanced Lighting/ Select Advanced Lighting*, comprobar que está seleccionada y activada la opción *Radiosity*.

### Primera fase: calidad inicial

En la primera fase lo principal es comprobar que la intensidad de la iluminación es la que se considera adecuada, aunque el resultado sea simple.

1. En *Parámetros de procesamiento de radiosity*, fijar la calidad inicial en 75% (de 60% a 80% puede ser adecuado para empezar) y el número de iteraciones para toda la escena en 2 (de 0 a 2, 3 o 4 según

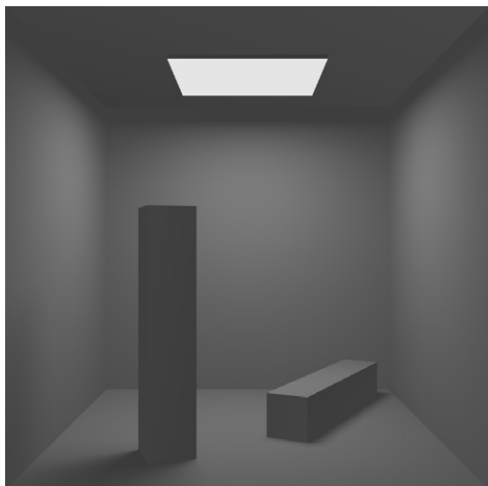


Figura 6.1 Radiosity. Configuración inicial: calidad 75%, 2 iteraciones



los casos). Presionar el botón *Start*. Con esto se inicia el proceso que puede ser más o menos lento según la complejidad de la escena.

2. Ajustar los parámetros y reiniciar el proceso hasta acercarse a la calidad deseada. No es recomendable fijar la calidad inicial por encima de 95%. Y 85% será suficiente en la mayoría de los casos.
3. Ajustar las iteraciones de refinamiento (esto sólo merece la pena si hay objetos visibles que sólo pueden recibir luz indirectamente). El grado de *filtración* puede fijarse en 1 o 2 una vez obtenidos resultados adecuados. Esto elimina errores pero también desenfoca ligeramente la imagen. El filtrado se realiza directamente en el visor, sobre el resultado de la radiosidad; no es necesario rehacer el cálculo, es un efecto de postproducción. El botón *Setup* da acceso directo al Control de Exposición que se describe más adelante.

Hay tres parámetros que pueden resultar equívocos por lo que no está de más algún comentario adicional. El contador *Refine Iterations (All Objects)* (Iteraciones de refinamiento para todos los objetos) especifica el número de iteraciones (de rebotes) que se aplican a la escena en conjunto. Conviene aumentar este valor en escenas en las que aparecen rincones visibles a los que no llega la luz. Pero no es necesario y puede retardar algo el cálculo si no es así. El contador *Refine Iterations (Selected Objects)* (Iteraciones de refinamiento para los objetos seleccionados) hace lo mismo pero sólo para los objetos que se hayan seleccionado antes de aplicar este contador. Si está claro que hay objetos críticos, por las razones mencionadas, esto puede ahorrar tiempo de cálculo. Cuando se restablece el cálculo de radiosidad la selección se mantiene si el contador está activado. La casilla *Process Refine Iterations Stored in Objects* (Procesar iteraciones de refinamiento incluidas en objetos) se refiere a propiedades específicas que se asignan a los objetos mediante el panel *Object Properties/ Advanced*

*Lighting* que se abre seleccionando un objeto y pulsando el botón derecho del ratón para entrar en sus propiedades. Cada vez que se refina una selección de objetos se aumenta el número de pasos guardados. Al reiniciarse el cálculo de radiosidad estos pasos se refinan automáticamente si está activada esta casilla. Esto se utiliza principalmente en animaciones para asegurar la homogeneidad de resultados en todos los fotogramas.

### Segunda fase: mallado

En la segunda fase se subdivide geométricamente la escena. También en este caso conviene proceder por fases. Un criterio adecuado es comenzar por dar al tamaño de la malla un 10% aproximado del de la escena, sin subdivisión jerárquica, y luego activar subdivisión jerárquica (adaptativa) con valores inferiores e inspeccionar el resultado.

La cantidad de subdivisión dependerá de los casos y habrá que inspeccionar los rincones con pequeños detalles para comprobar que se han subdividido adecuadamente. En la duda, comenzar por valores altos e irlos reduciendo. Tener en cuenta que un tamaño muy pequeño puede aumentar considerablemente el tiempo de cálculo.

En una escena como la del ejemplo las diferencias serán muy pequeñas. Hacer lo siguiente.

1. Aumentar la calidad inicial a 85%.
2. En *Radiosity Meshing Parameters* (parámetros de mallado de radiosidad) activar la casilla *Global Subdivision Settings* e introducir un valor adecuado para el tamaño de mallado (*Meshing Size*). En nuestro ejemplo el valor dado ha sido de 30. Con estos valores se obtienen los resultados de la primera figura adjunta.
3. En la misma sección, activar la casilla *Use Adaptive Subdivision* y establecer como valor máximo 12.0 (una tercera parte del anterior) y como valor mínimo 4.0 (una tercera parte del máximo). Repetir el cálculo así se obtendrán resultados como los de la



figura adjunta que muestran más detalle y manchas irregulares. Estas manchas desaparecerán en la fase final.

### Tercera fase: *regather*

En la fase final (sección *Rendering Parameters*) se refinan los resultados a escala de píxel, como ya he avanzado. Es decir que se realiza a la inversa: en función de la re-

solución de salida se calcula la contribución correspondiente a cada píxel y se ajustan los valores. Esto proporciona una calidad óptima y permite también jugar con valores geométricos más bajos, de modo que las diferencias se confíen al cálculo de recolección final. Hay que tener en cuenta de todas formas que la calidad de las sombras depende principalmente de la resolución de la malla y que esto no puede esquivarse. Aumentar esta resolución en puntos clave puede ahorrar tiempo, sobre todo en animaciones, porque sólo será necesario calcularlas una vez.

En esta sección hay otras opciones que explico brevemente. Se puede activar la opción *Re-Use Direct Illumination from Radiosity Solution* (reutilizar iluminación directa) si se quieren obtener sombras difusas de las luces de la escena. Con esta opción, las sombras se calculan a partir de la malla generada para el cálculo de radiosity. El resultado es más rápido que con la otra opción y las sombras resultan difusas (e imprecisas). Pero esto desactiva el cálculo de sombras asociado a las luces que haya en la escena por lo que habrá que elegir entre una u otra opción y, en la mayoría de los casos, será preferible dejar activada la opción predeterminada, *Renderizar iluminación directa* para que el cálculo de sombras se superponga al cálculo de radiosity.

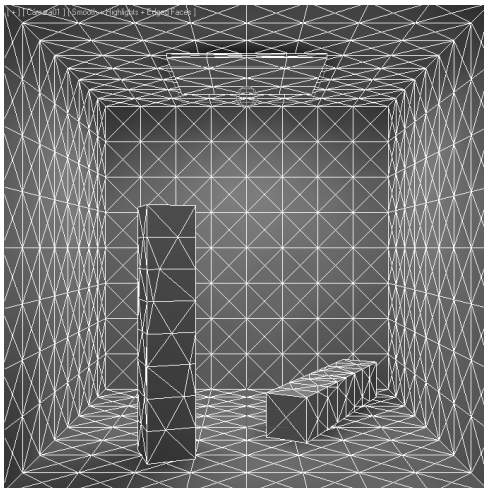


Figura 6.2 Radiosity. Malla regular de 30 cm

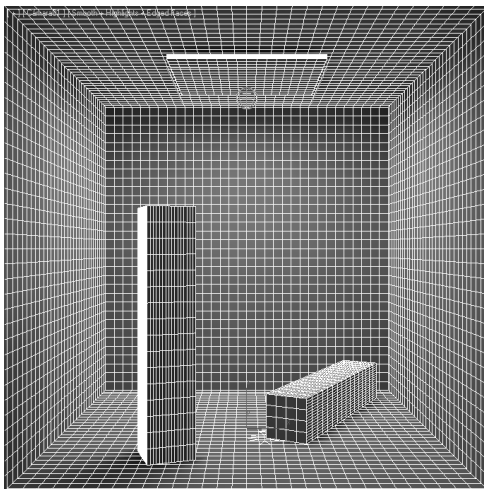


Figura 6.3 Radiosity. Malla adaptativo de 12/4 (max/min).

1. Una vez se haya llegado a resultados como los del apartado anterior, NO repetir el cálculo sino que pasar directamente a la siguiente sección.
2. Ir a *Rendering Parameters*, comprobar que está activada la opción *Render Indirect Illumination* y activar la casilla *Regather Indirect Illumination*.
3. Comprobar los parámetros (que se comentan más abajo) y modificarlos si se considera necesario. En la última figura adjunta se han dejado los valores por defecto (64/2.5/8x8 etc, *rays/ radius/ adaptive sampling*).
4. *Render*.



Las opciones disponibles en esta sección son: *Rays per sample* (rayos por muestra, valor inicial 64). Valores iniciales por debajo de 50 suelen dar resultados más bien toscos pero más rápidos y por encima de 150 bastante más exactos a costa de mayor tiempo; *Filter radius* (radio de filtro en píxeles, valor inicial 2.5). Si se aumenta, el resultado es más suave pero se puede perder definición; *Clamp values* (valores de bloqueo). Desactivado por defecto: si se activa, el valor inicial es 10.000 cd/m<sup>2</sup> que define el límite superior de luminancia a partir del cual se aplica la recolección. Suele utilizarse si aparecen puntos brillantes en la imagen, para eliminarlos. *Adaptive sampling* (muestreo adaptativo). En general es recomendable activarlo pues reduce el tiempo de cálculo haciendo que la recolección sea más intensa en zonas clave, según el grado de contraste detectado. Si se activa, las opciones disponibles son: *Initial Sample Spacing* (espaciado de muestras, 8×8 en versiones antiguas, 16×16 en las más recientes); *Subdivision Contrast* (contraste de subdivisión, 2.0 en versiones antiguas, 2.5 en las más recientes) que fija el nivel de contraste a partir del cual se incrementa el número de muestras; *Subdivide Down To* (subdividir hasta, 1×1 en versiones antiguas, 2×2 en las más recientes) que fija el espaciado mínimo para una subdivisión. Por último, *Show Samples* (mostrar muestras) hace que, si se activa, la imagen de salida incluye la distribución de las muestras como puntos rojos superpuestos a la imagen, lo que puede permitir juzgar mejor si los parámetros anteriores son adecuados o no.

### Control de las características y densidad de la malla

La subdivisión automática depende de la geometría previa. Si hay objetos con caras triangulares alargadas e irregularmente distribuidas, como ocurre cuando se hacen operaciones booleanas o se convierten contornos curvos a objetos sólidos por medio de una extrusión, la subdivisión puede reducir las

irregularidades pero seguirá siendo innecesariamente más densa en unas zonas que en otras. Por esta razón hay que procurar, si es posible (esto vale tanto para cálculos de radiosidad como para cálculos simples) que la geometría básica sea regular y no haya zonas mucho más grandes que otras o de formas muy diferentes. No olvidar que todos los cálculos de iluminación, con cualquier programa, se basan en una única normal que representa una cara como si estuviera concentrada en un punto. Por esto, aunque en el caso de radiosidad, esta observación debe tomarse como un principio prácticamente ineludible, en el caso de otros sistemas debe tomarse como una recomendación importante que no se debe perder de vista.

Si se utiliza la opción de subdivisión global este valor global puede variarse para objetos concretos. Para ello, seleccionar un objeto y editar sus propiedades de radiosidad (BDR/ *Object Properties/ Advanced Lighting*). Los parámetros disponibles en *Propiedades exclusivas de radiosidad*, permiten activar o desactivar opciones tales como arrojar sombras, recibir iluminación, computar las propiedades de reflexión, transmisión, etc. También se puede excluir del cálculo de recolección

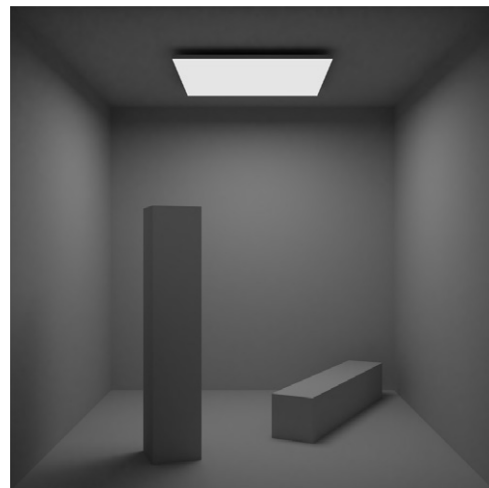


Figura 6.4 Radiosidad. Resultado final



(*regathering*). Si se desactiva la opción “Usar configuración global” quedarán disponibles las opciones de subdividir específicas para este objeto, con un tamaño de mallado concreto que es prioritario con respecto al dado por las opciones globales de radiosidad.

Otra opción, que proporciona un control aún mayor aunque es mejor no utilizarla si no se sabe bien como funciona, es utilizar el Modificador *Subdividir*. De hecho, lo que hacen las opciones de mallado de radiosidad no es ni más ni menos que aplicar automáticamente este modificador. Si se añade manualmente, aparece un cuadro de diálogo con un parámetro principal, *Size* que tiene un valor inicial aproximadamente igual al 10% de la dimensión mayor, idéntico a los de los cuadros de diálogo ya mencionados y que determina el tamaño de la subdivisión (corresponde aproximadamente a la longitud del lado del triángulo). Una vez que se ha especificado este valor, puede escogerse una de tres opciones para aplicarlo: a) *Automatic*: la densidad cambia automáticamente al modificar el valor, b) *Render*: la densidad cambia cuando se hace un *render*, c) *Manual*: la densidad cambia cuando, después de modificar el valor, se presiona el botón *Update Now* (actualizar ahora).

Y, en fin, otra alternativa es subdividir manualmente la malla mediante métodos de modelado del mallado en que aquí no podemos entrar pero que doy por conocidos. En cualquier caso no es una opción muy recomendable excepto en casos muy especiales.

### Control de la fuga de color con radiosidad

Al igual que ocurre con otros programas de iluminación avanzada, la interacción entre objetos da lugar a que las superficies de objetos cercanos a otros objetos con un color intenso, aparezcan teñidas de este color, un fenómeno que se denomina “fuga de color” o “sangrado de color” o “*color bleeding*”, en la literatura inglesa sobre estos temas.

Este fenómeno también se da en la realidad y por tanto es físicamente correcto. Sin

embargo nuestro sistema de percepción es extraordinariamente complejo y, entre otras cosas, tiene la capacidad de “descontar” éste y otros fenómenos a partir de una estimación de la iluminación global en relación con la reflectancia particular de las superficies, una capacidad denominada técnicamente “percepción de la constancia cromática” y que se intensifica en muchos casos con la intervención de la memoria, que también “corrige” el color percibido a partir del reconocimiento del objeto percibido.

Por esta razón a menudo conviene corregir este efecto. Si se utiliza *Final Gather*, se pueden utilizar *shaders* especiales que dan pleno control sobre esta corrección y que se explican en el libro sobre Simulación de Materiales. En el caso de Radiosidad hay que utilizar un material especial, específico para cálculos de radiosidad que, entre otras cosas, incluye controles para la fuga de color.

Este material se denomina *Architectural* (pues antes de la inclusión del material *Arch&Design* era el más adecuado para aplicaciones arquitectónicas) y sus propiedades sólo se aplican, en 3ds Max, con *Radiosity* o *Light Tracer*. No se puede aplicar con *mental ray*, y tampoco tendría sentido pues es preferible utilizar el nuevo material *Arch&Design*. Se utiliza por dos razones principales: a) para ajustar las propiedades de un material a los cálculos de estos dos sistemas de iluminación y para, entre otras cosas, controlar la fuga de color, b) para crear efectos especiales, fundamentalmente crear objetos autoiluminados y que aporten energía a la solución de radiosidad.

Si no se quiere pasar por tener que substituir un material ya creado por este otro tipo también puede utilizarse otro material, denominado *Advanced Lighting Override Material* (material de substitución de iluminación avanzada). En este caso, presionar el botón de tipo de material, seleccionar este material y en el cuadro de diálogo que aparecerá *Discard old material* o *Keep old material as sub-material*. Escoger la segunda opción. De este modo las características del material



que se estaba utilizando (color, reflectancia, texturas, etc) se mantendrán y los parámetros específicos del material *Architectural* se añadirán a los anteriores.

Estos parámetros complementarios (del material *Architectural* o de la extensión citada en el párrafo anterior) son los siguientes:

- *Reflectance Scale* (Escala de Reflectancia). Permite especificar el porcentaje de luz que refleja un material. Conviene recordar que los materiales corrientes tienen valores que oscilan entre un 20% como mínimo y un 70% como máximo. Sólo un cuerpo negro ideal refleja un 0% y sólo un espejo perfecto refleja un 100%.
- *Color Bleed* (Fuga de color). Afecta a la cantidad de color reflejado.
- *Transmittance Scale* (Escala de Transmisión). Permite especificar el porcentaje de luz que se transmite como transparencia o como traslucidez.
- *Luminance Scale* (Escala de Luminancia). Si es mayor que 0.0 afecta a materiales Estándar con Autoiluminación, que se hayan combinado con el Material de Substitución o bien a materiales Arquitectónicos que cuentan con un parámetro específico: “Emitir energía” (según luminancia) que se puede activar o desactivar. En cualquiera de los dos casos, el valor de luminancia controla la cantidad de luz que emite el material.
- *Indirect Light Bump Scale* (Relieve de luz indirecta). Se utiliza para escalar el efecto del mapeado de relieve lo que puede permitir corregir imprecisiones del efecto de relieve que afectan especialmente a las áreas iluminadas con luz indirecta.

Para controlar la fuga de color hay que utilizar los dos primeros. Con *Reflectance Scale* se controla la cantidad de luz reflejada. Y con *Color Bleed*, la saturación. En la mayoría de los casos bastará con ajustar los valores de *Color Bleed*.

## Radiosidad con otros programas

El programa más utilizado en aplicaciones arquitectónicas o similares, basado en radiosidad, es, probablemente, *Artlantis*.

*Artlantis* utiliza una forma simplificada de radiosidad, optimizada para su motor de render (*render engine*) que permite el procesamiento en tiempo real. De acuerdo con la filosofía del programa que es proporcionar la máxima simplicidad de uso, no hay parámetros complejos que haya que ajustar, como en el caso de 3ds Max u otros programas sino que todo se reduce a poco más que escoger una calidad “baja”, “media” o “alta”.

No hay, por tanto, mucho que añadir. Como es lógico, las posibilidades son más limitadas, sobre todo por lo que hace a la simulación de escenas interiores. Pero para un uso simplificado, principalmente en exteriores, que permita obtener resultados de cierta calidad con el mínimo esfuerzo de aprendizaje, es una buena opción.

Hay otros programas que utilizan radiosidad como *Lightwave 3D*, un programa potente y que incluye varias opciones de cálculo, entre ellas, métodos mixtos que combinan radiosidad con otros métodos basados en *Montecarlo* y *Final Gather*.

Otro programa, muy famoso en ciertas escuelas de arquitectura de vanguardia de Estados Unidos, aunque progresivamente menos utilizado, y que también utilizaba radiosidad como método principal, es *Form Z*.

Sin embargo, como ya he dicho, la mayoría de los profesionales se han decantado por otros métodos debido a las limitaciones intrínsecas de los métodos basados en radiosidad.



### 6.3 *Path tracing* y otros métodos básicos de Montecarlo

#### Métodos de *path tracing* con *iray* de *mental ray*

##### Descripción general

Desde la versión 2012 de 3ds Max es posible utilizar un motor de *render* alternativo denominado *iray*. Este módulo comenzó a desarrollarse, por *mental images*, en 2008 y en 2009 apareció la versión 1.0. Al año siguiente aparecieron versiones para varias aplicaciones, entre ellas *3ds Max* o *Catia*. La versión 2012 de *3ds Max* incorporaba la versión 1.2 de *iray*.

Aunque no hay demasiada información técnica sobre este motor parece claro que es una variante de *path tracing*: la ayuda dice literalmente que “crea representaciones precisas por medio del trazado de light paths” y también se añade que en futuras versiones podrá incorporar *bi-directional path tracing* (BDPT). La información complementaria que se puede encontrar en Internet explica que está basado en QMC (*Quasi Monte Carlo*) aunque no he encontrado información sobre las variantes concretas. En cualquier caso, como ya hemos visto en capítulos anteriores, el uso de muestras no estrictamente aleatorias reduce hasta cierto punto el ruido pero a costa de crear otros defectos que según los casos pueden ser más o menos perceptibles en forma de patrones recurrentes. Por otra parte la cantidad de ruido no decrece linealmente sino que permanece un remanente que es difícil de eliminar si no se acepta que se disparen los tiempos o se recurre a métodos complementarios, tales como la utilización de filtros que desenfocan el resultado.

Un problema importante que afecta a la inclusión de *path tracing* en las aplicaciones actuales es, como también he comentado, su lentitud y que las tarjetas gráficas corrientes en 2012 no cuentan con recursos para acelerarlos. La inclusión de *iray* en las últimas versiones de algunos programas conocidos ha sido posible porque Nvidia se ha involucrado

en este proyecto. Esto quiere decir que para poder utilizarlo se requiere una buena tarjeta gráfica, más concretamente, una GPU que incorpore la tecnología CUDA (ver el capítulo 4), y que cuente con el mayor número posible de núcleos (*CUDA cores*) pues *iray* está optimizado para trabajar con esta tecnología.

Tiene algunas limitaciones y peculiaridades que resumo a continuación. Procesa bien la mayoría de los algoritmos de representación más o menos sofisticados, tales como desplazamientos, recortes (*cutouts*), autoiluminación o HDR. Además, dado que un motor de *render* basado en trazado de rayos no distingue entre tipos de rayos “de luces”, “de reflexión” o “de sombras”, la configuración de luces y materiales se simplifica considerablemente en muchos casos. Es particularmente efectivo para procesar reflexiones, tanto especulares como difusas. También es muy efectivo, más que otros motores y métodos, para representar objetos con autoiluminación que pueden utilizarse como alternativa a luces virtuales. Y, más aún, hay efectos como la simulación de la profundidad de campo (DOF, *Depth Of Field*) que se dan por añadidura sin necesidad de utilizar recursos sofisticados complementarios y sin grandes diferencias en el tiempo de cálculo. También permite incluir relieves virtuales por medio de *bumps*, en iluminación indirecta, sin que aparezcan defectos notorios como en otros algoritmos. En fin, una ventaja también importante de este motor es que es relativamente independiente de la complejidad de la escena y que calcula bien la iluminación de formas complejas, con mayor precisión que otros motores.

Por otro lado, sí es dependiente de los recorridos de los rayos de luz y tarda más en procesar escenas que incluyan muchos rebotes o desviaciones de los rayos principales. Una escena con muchos rincones a los que la luz llegue después de múltiples rebotes tardará más tiempo en procesarse que una escena que muestre un escenario abierto, con pocos rincones.





La complejidad de los **materiales** también puede afectar al tiempo del proceso y hay algunos tipos que no se procesan. Cuanto mayor sea la complejidad de las propiedades de los materiales, cuantas más texturas o combinaciones de parámetros se incluyan, mayor será también el tiempo de cálculo. También hay que tener en cuenta, aunque no sea demasiado corriente, que no puede procesar *shaders* programables como hace *mental ray*. Por otro lado, aunque el material *Arch&Design* se procesa bien, hay algunas características especiales de este material que no se tendrán en cuenta aunque sí se procesarán en algún caso: a) las correspondientes a los parámetros básicos de *Reflection* (*glossy samples*, *fast interpolate*, *highlights+fg*) que corren a cargo de *iray*, b) las correspondientes a *Ambient Occlusion* y *Round Corners* (en la sección *Special Effects*) c) las que incluyen parámetros que dependen de *Final Gather*, d) cualquier otra que dependa de los *shaders* que aparecen en la sección “mental ray connection”. No procesa materiales SSS lo que no es muy grave para aplicaciones arquitectónicas, aunque hay algunos usuarios que sacan un gran partido de las sutilezas de este material para simular mármol o alabastro o materiales especiales, no necesariamente piel humana.

Se puede descargar un material específico, *iray material*, de *mental images*. No doy la url porque puede haber cambiado pero bastará hacer una búsqueda con las palabras claves “iray”, “material” “mental ray”. Ver el libro de Simulación de Materiales para más información sobre este material.

En el caso de mapas de bits, debe tenerse también en cuenta que hay algunas limitaciones aunque no demasiado importantes. Consultar la ayuda si aparece algún problema imprevisto (en *Renderers / iray*).

En caso de que un material no se procesase, la superficie correspondiente aparecerá de color gris y se emitirá un mensaje a través de la *Render Message Window*.

Tampoco se procesan otras utilidades como *Render to Texture*, *Panorama Exporter*, *Effects*, *Video Post* o *Lighting Analysis*.

Por lo que respecta a los **tipos de luces** una limitación no importante, pero que debe tenerse en cuenta, es que las luces deben ser fotométricas o de algunos de los tipos de mental ray asimilables a éstas, concretamente *mr Sun* y *mr Sky*, además de *mr Sky Portal* (que ya se incluye con los tipos fotométricos). Si se utilizan luces simples no se procesarán. Y utilizar luces fotométricas implica también, como ya hemos visto, que hay que activar el control de exposición, que debe ser el *mr photographic exposure control*.

El uso de *sky portals* es importante, como ya hemos visto, para que la iluminación de un interior desde el exterior sea más efectiva pero en *iray* no mejoran de modo apreciable el resultado e incluso pueden dar lugar a algún problema si el sol entra directamente por una ventana. Pero ya hemos visto que esto también puede ocurrir con los procedimientos corrientes de *Final Gather* (y que una solución efectiva en este caso es reducir o anular la intensidad del disco solar, lo que no afecta a la iluminación global). En cualquier caso no parece haber demasiadas razones para utilizarlo dado que, entre otras cosas, la distribución de la iluminación con *iray* es más efectiva.

Como ya he dicho el cálculo de **sombras** corre a cargo del propio motor de *render* por lo que las especificaciones de parámetros de sombra no se tienen en cuenta. En el caso de *mr Sun* tampoco se tienen en cuenta las especificaciones correspondientes a *Sun Photons* y *Aerial Perspective*. En el caso de *mr Sky*, el modelo de cielo debe ser *Haze*.

El motor *iray* procesa muy bien los **objetos con autoiluminación** como ya he dicho más arriba. De hecho puede procesarse una escena sin luces, sólo con objetos que emiten luz, objetos a los que se haya aplicado un material con estas características. Para procesar una escena con un objeto autoiluminante, en *3ds Max*, añadir previamente una luz cualquiera a la escena y luego apagarla. De este modo se desactivan las luces predeterminadas que *3ds Max* añade automáticamente a la escena sin no hay ninguna luz activa. Luego crear un objeto adecuado y añadirle un material con





estas características. La mejor opción será utilizar el material *A&D* y activar los parámetros correspondientes a la sección *Self-Illumination*. En la descripción del ejemplo que sigue se indican los parámetros adecuados para la escena que se ilustra.

### Parámetros

A diferencia de otros sistemas, la configuración es mínima lo que sin duda supondrá un importante atractivo para usuarios que no quieren complicaciones técnicas. Todo lo que hay que hacer es escoger una de tres opciones para que el proceso se detenga: el tiempo, el número de iteraciones o la intervención del usuario para detener el proceso.

Para escoger este motor, en *3ds Max*, ir a *Render Setup/Common* y, en la sección inferior, *Assign Renderer*, descolgar la lista junto a *Production* y seleccionar *iray*. Al hacer esto, el *Render Setup* cambiará y solo mostrará dos secciones. Los parámetros principales se encuentran en la primera sección y son los siguientes:

- a) *Time*. La duración predeterminada es de 1 minuto. Puede asignarse cualquier duración aunque este tipo de especificación solo será práctica cuando se cuenta con una cierta experiencia.
- b) *Iterations*. Especifica el número de iteraciones o pases. El valor predeterminado es 500, un valor relativamente alto que convendrá rebajar para pruebas.
- c) *Unlimited*. El proceso continúa indefinidamente hasta que se cancela por el usuario.

En la segunda sección, *Advanced Parameters*, se pueden especificar los siguientes parámetros complementarios:

El grupo *Trace/Bounce Limits* (Límite de trazado de rayos y rebotes) incluye los siguientes. *Physically Correct* (sin límite) y *Maximum Number of Light Bounces* (el valor predeterminado es 4 rebotes y el mínimo 2). Esta última opción es adecuada para hacer pruebas previas pero debe tenerse en cuen-

ta que el resultado será más oscuro, sobre todo en escenas interiores con muchos rincones que aparecerán negros o casi negros si se limita este valor. Ver la figura adjunta que muestra un ejemplo de las diferencias que se obtienen con diferente número de rebotes.

El grupo *Image Filtering (Antialiasing)* permite elegir entre tres tipos de filtros: *Gauss* (predeterminado), *Box* y *Triangle*. Ver el capítulo 4 para más información sobre estos tipos de filtros de *antialiasing*.

El grupo *Displacement* solo tiene sentido si se utilizan materiales con desplazamiento. Los parámetros son similares a los que se usan con *mental ray*. Pero los parámetros por defecto para *Edge Length* (2.0) y *Max Displacement* (20.0) son muy altos y convendrá bajarlos. Ver el libro de Simulación de Materiales para más información sobre este *shader* de relieve virtual y estos dos parámetros, además del límite de *Max Subdivs* (16K) y la opción de suavizar o no el resultado (*Smoothing*).

El filtrado de texturas no es necesario pues está incorporado de modo automático al tipo de procesamiento de *iray* dado que este efectúa un supermuestreo de modo implícito al enviar cientos de rayos por cada píxel.

El grupo *Material Override* funciona igual que en *mental ray* (se usa para recubrir toda la escena con un único material, lo que resulta particularmente práctico para hacer pruebas de iluminación sin que las propiedades de los materiales interfieran en el resultado, como ya hemos visto).

Al iniciar el proceso la escena se va representando con refinamientos progresivos. Las primeras imágenes son muy granuladas y, a medida que aumenta el número de iteraciones, el grano se va reduciendo. Las figuras adjuntas muestran estadios característicos de este proceso.

### Ejemplo 1 (interior)

La escena de las figuras adjuntas tiene el mismo aspecto que las del resto de este capítulo. Pero hay algunas diferencias importantes que tienen que ver, por un lado, con algunas limi-

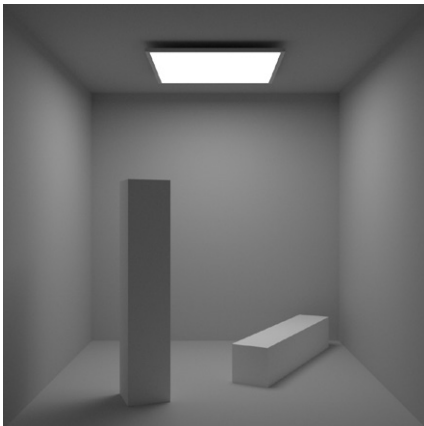
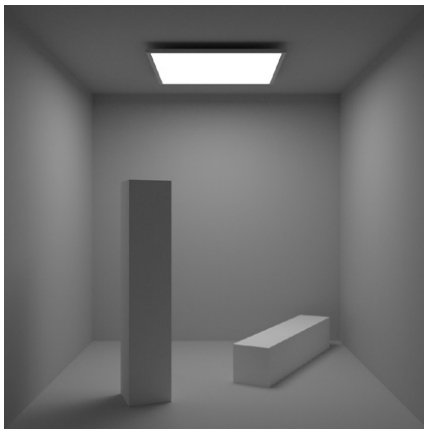
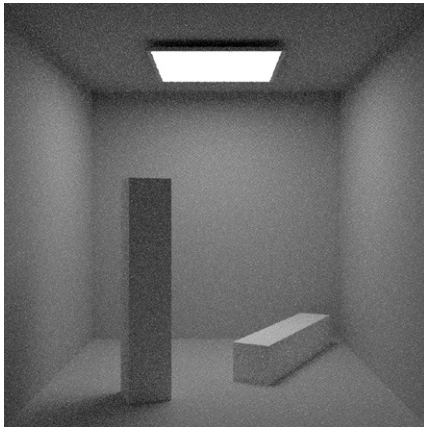


Figura 6.5 iray. Ejemplo 1 configurado con: a) 5 iteraciones, tiempo 1X (14"), b) 50 iteraciones, tiempo 7.5X, c) 500 iteraciones, tiempo 120X

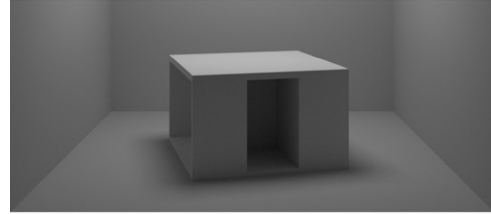
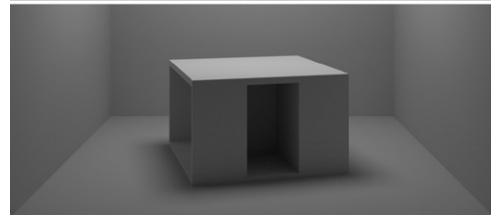
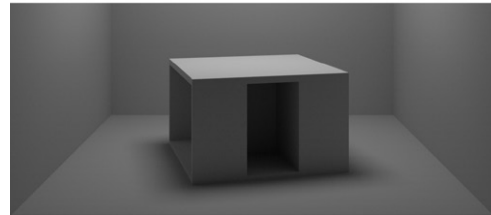
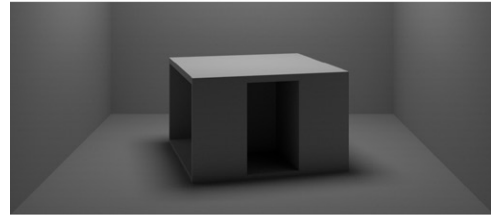
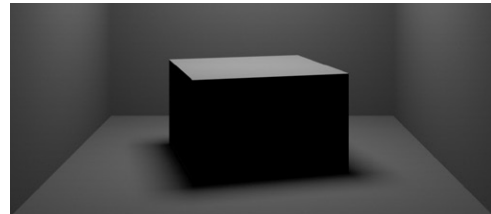


Figura 6.6 iray con 500 iteraciones y: a) 2 rebotes, tiempo 1X, b) 3 rebotes, tiempo 1.9X, c) 4 rebotes, tiempo 2.8X, d) 6 rebotes, tiempo 4.4X, e) 9 rebotes, tiempo 5.3X



taciones de *iray* y, por otro, con alguna de sus ventajas.

Como los *clipping planes* de la cámara no funcionaban he utilizado otro recurso similar. He alargado el recinto para que la cámara quede dentro y he colocado un plano de 3x3 metros en la posición en la que estaba el plano frontal (a -150 cm del centro del recinto). En *Object Properties/General/Rendering Control*, he desmarcado la propiedad “Renderable” de este plano. De este modo el plano contribuye a los reflejos e iluminación de la escena pero no aparece en la representación final.

Por otro lado, en lugar de una luz fotométrica he utilizado un objeto emisor. Este objeto tiene asignado un material de tipo *mr Arch&Design* con *Self-Illumination* activado. La luminancia de autoiluminación se ha establecido en 2.000 cd. El Control de Exposición es de tipo *mr Photographic* con EV 8.0.

Como ya he dicho más arriba, la configuración de cálculo es muy sencilla y se limita a ajustar el número de iteraciones. A medida que aumenta este número, el resultado se va refinando y el ruido disminuye. Las tres figuras del primer grupo muestran los resultados con diferente número de iteraciones.

Uno de los principales ajustes es que se pueden hacer, y que afecta de modo importante al resultado en determinadas situaciones, es el número de rebotes.

En el *Render Setup* de *iray*, en la sección *Renderer / Advanced Parameters*, desactivar la opción predeterminada, *Physically Correct (Unlimited)* y activar la opción *Maximum Number of Light Bounces* (número máximo de rebotes de luz) que incluye un contador. De este modo se pueden limitar los rebotes, según lo que se especifique en este contador lo que servirá también para acelerar el cálculo, principalmente en las fases de pruebas, aunque los resultados no sean correctos físicamente. Las figuras adjuntas, con un objeto (una escultura de Donald Judd) con rincones a los que no llega fácilmente la luz, muestran las diferencias que se pueden dar con diferente número de rebotes.

## Ejemplo 2 (exterior)

Preparar una escena sencilla como la de la figura adjunta que consiste en los mismos objetos que en las escenas interiores pero colocadas sobre un plano.

Crear un sistema de luz diurna. Ajustarlo manualmente para que las sombras sean similares a las de la figura adjunta y seleccionar *mr Sun* y *mr Skylight* como tipos de luz solar y celeste. Comprobar el Control de Exposición

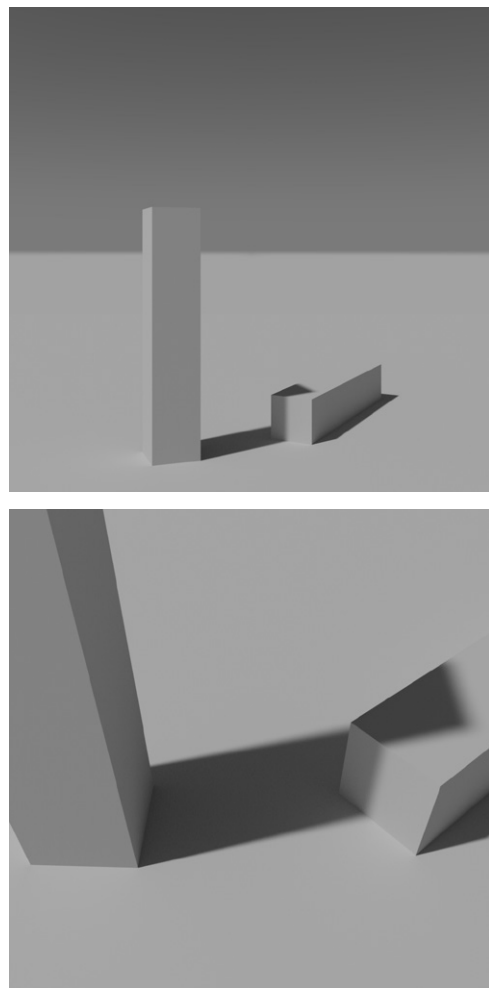


Figura 6.7 *iray*. Ejemplo 2: a) Vista general, b) Vista de detalle



que se deberá haber ajustado automáticamente a *mr Photographic* con EV 15.0.

Seleccionar *iray* como motor de *render*. Ajustarlo con un número suficiente de iteraciones. En el ejemplo adjunto se ha utilizado el valor predeterminado, 500. Si utilizáramos diferentes valores obtendríamos variantes como las ilustradas en el ejemplo anterior.

Eso es todo. En la gran mayoría de los casos no hará falta hacer mucho más.

### **Path tracing con V-Ray (light cache)**

#### **Procedimientos de configuración de iluminación global en V-Ray**

V-Ray utiliza cuatro sistemas de cálculo: *Irradiance*, *Photons*, *Brute force GI* (DMC, QMC) y *Light cache*. Los dos primeros se describirán en los apartados correspondientes, más adelante. Los otros dos se discuten a continuación.

La configuración se lleva a cabo, en cualquier caso, activando el comando general *Options*. Así se despliega un panel con varias secciones (*Global Switches*, *Image Sampler*, *Caustics*, *System*, *Displacement*, *Environment*, *Camera*, *VFB Channels*, *Output*, *Indirect Illumination*, *DMC Sampler*, *Color Mapping*...). La que nos importa principalmente para lo que sigue es *Indirect Illumination*. Esta sección es la principal para la configuración del cálculo de iluminación. Incorpora los siguientes grupos principales: a) *GI (Global Illumination)*. Incluye un solo parámetro para activar o desactivar la iluminación global y dos opciones para activar *Refractive GI Caustics* y *Reflective GI Caustics* que sólo tienen efecto si hay materiales preparados para generar caústicas y se han activado los parámetros requeridos (ver más abajo) y si estos materiales reflejan o refractan la luz, b) *Post-Processing* incluye algunos parámetros para ajustar algunos valores del resultado final, como la saturación o el contraste, c) *Primary Engine* y d) *Secondary Engine*, que permiten escoger el motor primario y secundario de cálculo. Son las dos elecciones principales que hay

que hacer antes de ajustar la configuración general. En función de lo que se escoja en este grupo cambiarán las siguientes secciones para mostrar los parámetros propios de cada método. Las cuatro opciones disponibles son las correspondientes a los cuatro sistemas de cálculo citados al comienzo de este apartado. La *Secondary Engine* cuenta además con una opción adicional, *None*, para desactivar el segundo motor. Cada uno de estos grupos cuenta con un *Multiplier*, cuyo valor predeterminado es 1.0 en el primer caso y 0.85 en el segundo. El primero afecta a la contribución de los rebotes primarios. No es conveniente modificarlo pues distorsionaría la luminosidad general. El *Multiplier* para el cálculo secundario (*secondary bounces*) afecta a la iluminación de los rebotes. Aumentar este valor es un recurso habitual para conseguir que las zonas en sombra, a las que no llega directamente la luz queden más iluminadas. Sin embargo esto afecta también a la fuga de color (*bleeding*) de una superficie a otra por lo que no suele ser conveniente subirlo demasiado.

En general, en V-Ray hay tres niveles de cálculo que deben tenerse en cuenta, tanto para este método como para los que siguen:

a) Luz directa (*Direct Light*). El cálculo se hace directamente a partir de la posición y la intensidad de las luces presentes en la escena. Es el punto de partida para los cálculos de iluminación indirecta y se procesa de modo independiente.

b) Derivaciones o rebotes primarios (*Primary Bounces*). El cálculo se basa en los rayos emitidos por una superficie a la que ha alcanzado un rayo de luz primaria. Estos rayos son los que tienen mayor influencia en el cálculo de GI. Si sólo se activa el motor primario de cálculo son los únicos rayos que se tienen en cuenta.

c) Derivaciones o rebotes secundarios (*Secondary Bounces*). El cálculo se basa en los rayos emitidos por otras superficies a las que hayan alcanzado los rebotes primarios y los rebotes sucesivos derivados de estos que irán amortiguándose en función de factores



diversos: la reflectancia de los materiales, su orientación y distancia con respecto a otras superficies, etc. En el caso de escenas externas su importancia es mucho menor pues la mayoría no llegan a alcanzar otras superficies y se desvanecen en la distancia.

### § § §

El **antialiasing** general (bordes escalonados, etc) se controla globalmente desde la segunda sección de *Options*, la sección *Image Sampler*. Esta sección incluye tres alternativas: *Fixed rate* (toma un número fijo de muestras por píxel), *Adaptive DMC* (Muestreo Adaptativo de *Deterministic Monte Carlo* que toma una serie de muestras variable por píxel basándose en la diferencia de intensidad entre el píxel y sus vecinos) y *Adaptive subdivisions* (aumenta la subdivisión en las zonas en que se detecta mayor contraste (en general, bordes de objetos).

Por lo general es preferible la tercera opción que incluye parámetros muy similares a los que ya hemos visto para *mental ray*. Los dos principales son *Min / Max rate*. Con *Min rate* se especifica el mínimo número de muestras por píxel: 0 significa 1 muestra por píxel, -1 una muestra cada dos píxels, -2 una cada dos píxels, etc. Con *Max rate* se especifica el máximo número de muestras por píxel: 0 (1 muestra por píxel), 1 (4 muestras), 2 (8 muestras, etc). Los valores por defecto *Min / Max* -1/2 suelen ser suficientes. Rebajarlos a -2, -1 para pruebas rápidas o subirlos a 0,2 para aumentar la calidad.

Los demás parámetros son *Threshold* (0.1), que controla el umbral de diferencias de intensidad: si se baja el valor añadirá más muestras con cambios de intensidad menores y viceversa; *Rand* añade algo de ruido aleatorio a las zonas suavizadas lo que ayuda a esconder defectos en zonas críticas; *Object outline* hace que se supermuestreen los bordes de los objetos en cualquier caso.

El grupo *Antialiasing Filter* permite escoger un filtro. Las opciones son: *Area*, *Box*, *Triangle*, *Catmull Rom* (más definición pero riesgo

de efectos de Moiré), *Lanczos* y *Sinc*. Ver la sección sobre filtros del capítulo 4, donde se da más información sobre estos filtros.

Aparte de estas configuraciones generales hay un gran número de controles para ajustar la configuración específica de cada motor de cálculo que resumiré en cada caso.

### § § §

Lo que sigue de esta sección está basado en elegir la opción *Light Cache*. La versión 2.0 incluye soporte para *bidirectional light trace* pero no se detalla.

Como decía al comienzo de esta capítulo, la escena utilizada para los ejemplos con SketchUp es idéntica a la utilizada con *mental ray* con las excepciones menores que se han mencionado.

## Descripción general

El método que se describe en este apartado está desarrollado específicamente para este programa, si bien su descripción coincide con el método general de *path tracing* que ya hemos visto. Hay múltiples posibilidades técnicas de concretar este método general y en V-Ray se denomina *light map* o *light cache* ("captura" de la luz) e incorpora una serie de recursos que complementan el método básico.

Se presenta en dos variantes principales (ver el apartado *Mode*, en los parámetros generales, del interface del programa): *Single Frame* y *Progressive Path Tracing*. El primer modo es el más adecuado para utilizar *Light Trace* como motor secundario (por ejemplo con *Irradiance*). El segundo es más adecuado si se quiere utilizar *Light Trace* como motor primario y, en este caso, una buena opción es utilizar este mismo método y modo como secundario.

Este método básico, como también hemos visto, se basa en construir un mapa (*light cache* en la variante de V-Ray) por medio de varios recorridos, *eye paths*, que se inician en el punto de vista y van almacenando los



valores de iluminación en cada punto de intersección con las superficies que encuentran en su camino. Así se crea una estructura 3D muy parecida a la de los mapas de fotones. La principal diferencia con los mapas de fotones es que el mapa se construye a partir de la cámara mientras que el de fotones se construye a partir de la luz.

Tiene por tanto una serie de ventajas y desventajas características. Al iniciarse en la cámara es dependiente del punto de vista. Pero por esta misma razón puede optimizar el tamaño y el rastreo adaptándolos a lo que importa y descartando zonas que no serán visibles o tendrán una repercusión insignificante sobre el resultado. Y solo requiere construir un trazado 3D, a diferencia de los métodos basados en enviar rayos desde la luz que, en principio, deben repetir esta construcción para cada luz de la escena. También permite trabajar con cualquier tipo de luz. Otra ventaja importante es la velocidad, al igual que ocurre con los mapas de fotones. Y, como éstos, funciona mejor combinado con otros métodos, como irradiancia, para eliminar el ruido residual. Otras ventajas que cabe citar es que reconoce todo tipo de luces o que reproduce mejor los efectos de atenuación en esquinas. Y entre las desventajas que conviene no olvidar están que es dependiente de la vista (al igual que los mapas de irradiancia) lo que, pese a las ventajas citadas, obliga a repetir el cálculo para cada cámara, que depende de la resolución y por tanto es difícilmente adaptativa o que no reconoce bien los mapas de relieve (*bump maps*). Por estas razones, al igual que el mapa de fotones, es más adecuado para utilizar como segundo motor de render. No tiene un parámetro para controlar el número de rebotes pues este número es teóricamente infinito con este método: el cálculo se detiene en un determinado punto y hasta llegar a ese punto los rayos siguen rebotando por toda la escena. La limitación viene dada por las propiedades del material o por lo especificado en *Global Switches*.

Esto hace que el resultado sea algo más luminoso que con otros métodos. Si,

por ejemplo, se calcula una escena con *Irradiancia+QMC* con un número determinado de rebotes) y luego con *Irradiancia+Light Cache*, la segunda resultará ligeramente más iluminada debido al mayor número de rebotes.

### Parámetros generales

Para acceder a la configuración de este método, ir, en el panel principal de V-Ray, a *Options/Indirect Illumination* y, para repetir el análisis que sigue, seleccionarlo como motor primario (*primary engine*). Desactivar el motor secundario (escogiendo *None*) o escoger también *light cache* como secundario. Así nos concentraremos en las características de este método. Los parámetros generales se resumen a continuación.

El grupo *Calculation parameters* incluye los siguientes. *Subdivs* (800), que determina el número de *paths* trazados desde la cámara y es el parámetro principal que se comentará con más detalle más abajo. *Sample size* (0.02) determina el espaciado de los rayos en el *cache* de la luz y también afecta de un modo importante al resultado, por lo que le dedicaré más espacio en el apartado de pa-

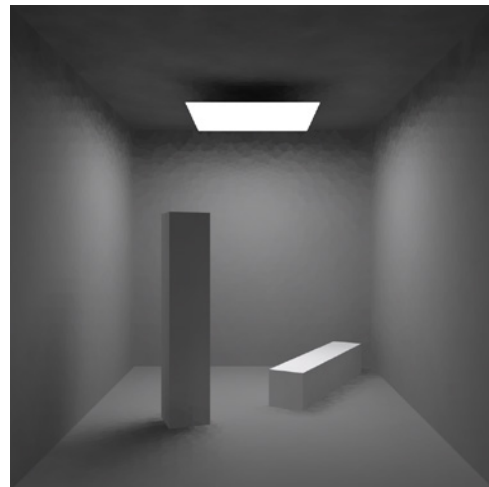


Figura 6.8 V-Ray con Light Cache. Valores predeterminados 800 / 0.02 (subdivisiones / sample size)





rámetros principales. Otro tanto cabe decir de *Scale (Screen)* que determina las unidades en que se basa el parámetro anterior y también se comenta de modo más amplio más abajo. *Passes* (4) establece el número de pasos que se combinan para el cálculo final. La modificación de este parámetro puede ser importante en el caso de *rendering* distribuido pero no

hace falta tenerlo en cuenta en un sólo ordenador. *Depth* (100) puede servir para limitar la longitud de los rayos trazados aunque en la práctica esto puede no tener efecto debido al carácter recursivo del algoritmo. *Min Paths per sample* (16) fijaría el número mínimo de recorridos que se trazan por cada muestra y no es necesario, sin una buena razón, cam-

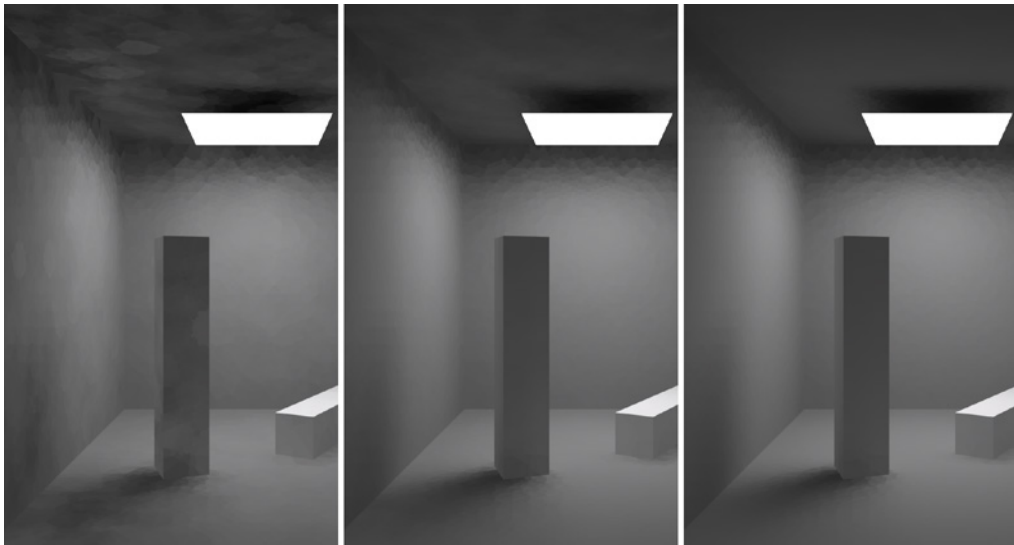


Figura 6.9 V-Ray con Light Cache. Modificación de las subdivisiones (sample size 0.02): a) 50, b) 100, c) 500, d) 1000

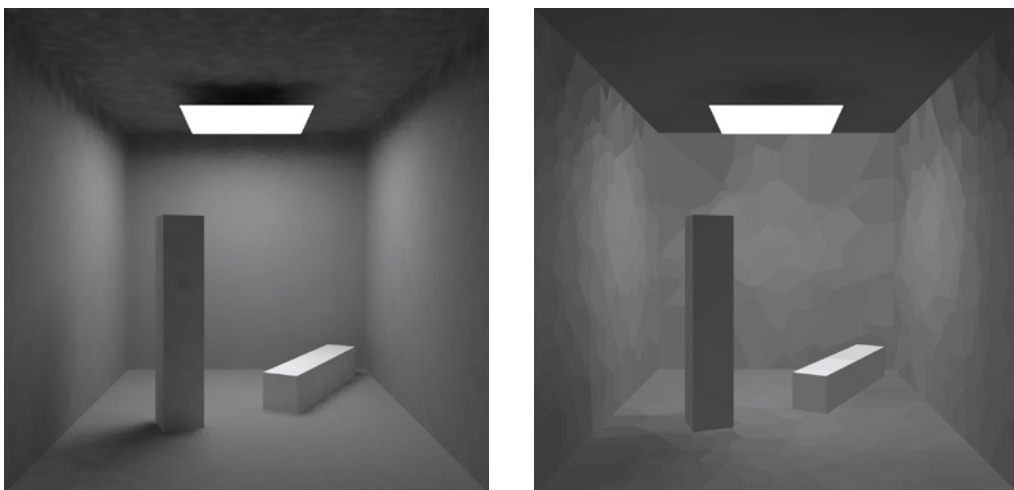


Figura 6.10 V-Ray con Light Cache. Modificación del sample size (subdivisiones 800): a) 0.01, b) 0.1



biar el valor predeterminado. *Store direct light* (act) almacena los valores de luz directa si se mantiene activada, que se suman al cálculo; si se desactiva, se deducen del cálculo global, una opción que puede ser preferible en escenas con muchas luces y quizás también cuando se usa *light cache* como motor secundario. *Show calc phase* (desact) no afecta al cálculo pero sirve para ver las huellas de los recorridos calculados con lo que se puede evaluar mejor el proceso. *Adaptive sampling* (desact) se supone que puede mejorar la distribución añadiendo más muestras en las direcciones de donde viene la luz. Sobre *Adaptive Dirs Only* (desact) y *Multiple\_views* (desact) no he encontrado información, ni en la ayuda del programa ni en foros de Internet pero no parece que sean demasiado importantes.

El grupo *Reconstruction parameters* incluye parámetros que afectan principalmente al modo de suavizar los resultados de la interpolación del mapa. *Pre-filter* (desact) activa los prefiltros con el valor que sigue, *Prefilter Samples* (10) hace que las muestras se filtren antes de ser representadas, igualándolas con las vecinas. Cuanto mayor sea el número más se suavizará el ruido pero también puede perderse más detalle. *Use for Glossy Rays* (desact) hace que, si se activa, el mapa de *light cache* también se utilice para el cálculo de rayos brillantes lo que puede acelerar el cálculo en escenas con muchos brillos especulares. *Filter Size* (0.06) ajusta el tamaño del filtro seleccionado con *Filter Type* (*Nearest*), que incluye una lista colgante para elegir un tipo de filtro adecuado. Si se elige "None" la velocidad aumentará pero también el ruido lo que puede atenuarse con un valor adecuado de *Pre-filter*. Me remito de nuevo al apartado sobre filtros en el capítulo 4 de este libro.

El grupo *Mode* especifica el modo de construcción del mapa de *light cache*. La opción por defecto, *Single frame* (predeterminada) computa un nuevo mapa de *light cache* para cada imagen. Con *Progressive path tracing* el algoritmo se utiliza para muestrear la imagen final de modo progresivo. No hay diferencia

substancial entre estos dos modos por lo que hace a la calidad de los resultados si bien el sentido de los parámetros principales puede variar. *Fly-Through* se utiliza en animaciones partiendo de la base de que sólo la cámara cambia y reutilizando el mapa del primer marco. *From file* carga el *light cache* a partir de un archivo grabado previamente. El botón (...) junto a *File* se utiliza para grabar el resultado (con el botón *Save*, del grupo *Current Map*) en un fichero que puede reutilizarse.

Los grupos *Post render* y *Current Map* tienen el mismo significado y parámetros que en los otros métodos de cálculo. Otro tanto ocurre con algunos otros parámetros no demasiado importantes y que omito de esta descripción.

## Análisis

Los **parámetros principales** son los siguientes:

a) El principal de todos es *Subdivs* que, como ya he dicho, determina el número de *paths* trazados desde la cámara. El número real de recorridos es el cuadrado de este valor, es decir que 100 subdivisiones darían lugar a 10.000 rayos, 500 a 250.000 y 1000 subdivisiones darían lugar a 1 millón. El valor predeterminado es 800 (en las últimas versiones que he utilizado, 1.000 en versiones anteriores). Conviene hacer pruebas iniciales con valores del orden de las 100 o 200 subdivisiones y multiplicar por ocho o diez este valor para la salida final, es decir, 800 o 1000, para establecer configuraciones que darán buenos resultados aunque en algunos casos puede ser necesario llegar a los 2.000. Tener en cuenta que aumentar el número de subdivisiones aumenta el tiempo de modo exponencial: multiplicar por 2 el número de subdivisiones puede suponer multiplicar por 4 el tiempo de cálculo. En las últimas versiones que he utilizado (2013) el valor máximo admitido es de 60.000, un valor que carece de sentido práctico pues supondría  $(60.000^2)$  3.600.000.000 *paths*.



b) El valor de *Sample Size* también tiene importancia. Si es bajo, la densidad de rayos será mayor y los detalles quedarán mejor registrados. Sin embargo aumentará la ocupación de memoria y se generará más ruido. Si el valor es alto, la densidad será menor y los detalles quedarán peor registrados. Pero se utilizará menos memoria y el ruido será absorbido por la interpolación. La interpretación de los valores depende de si se ha escogido “Screen” (porcentaje de la resolución total en píxeles asociables a la cámara) o “World” (unidades reales de la escena). Un valor de 0.02 (*Screen*) significa que se tomarían muestras de un 2% del total de lo que ve la cámara (el máximo es 1.00 y el mínimo 0.0). Esto quiere decir que si la imagen tiene 1.000x1.000 píxeles estaremos tomando muestras en un entorno de 20x20 píxeles. En general, es preferible utilizar *Screen* pues nos da valores independientes de la resolución y bien adaptados al tipo de procesamiento que se lleva a cabo desde la cámara. Puede haber razones concretas para utilizar *World* pero debe tenerse en cuenta que en este caso las muestras serán independientes de la posición de la cámara pues el valor no es un porcentaje de un total sino un umbral absoluto. Esto puede ser conveniente en casos en los que hay muchos cambios de cámara pero poco indicado para vistas fijas. En los ejemplos que se adjuntan solo he utilizado la alternativa “Screen”.

Por lo que respecta al *antialiasing*, en general, si los parámetros generales, de *Image Sampler*, que ya he descrito más arriba, interfieren con los del método de GI utilizado son estos últimos los que tienen preferencia.

Con *light cache* el tipo de muestreo establecido desde *Image Sampler* (*Fixed*, *Adaptive*, etc) se ignora pues el algoritmo utilizado incorpora *supersampling* (varias muestras por píxel, como ya hemos visto en el capítulo 4). El tipo de filtro de *antialiasing*, sin embargo, sí se tiene en cuenta. Pero los filtros principales que conservan el detalle (Mitchell-Netravali, Catmull-Rom) pueden introducir más ruido. Y filtros de suavizado, como *Blend*, aumentarán

el tiempo de computación. Por lo que quizás es mejor desactivarlos.

Los defectos más característicos se pueden manifestar, en algunos casos, como fugas de luz en objetos de poco espesor o irregularidades en forma de manchas en zonas iluminadas por rebotes. Reducir el valor de *Sample Size* corrige estos efectos aunque consume más memoria.

## “Fuerza bruta” o DMC (*Deterministic Monte Carlo*) con V-Ray

### Descripción general

El método de “fuerza bruta” (*brute force*) se basa en recalcular los valores de GI, de modo independiente, para cada punto que será representado. Es un método muy lento pero muy exacto. Puede ser adecuado para escenas con pequeños detalles que no queden adecuadamente representados con otros métodos. Para acelerar el cálculo se suele combinar con mapas de fotones (*photon map*) o mapas de luz (*light map*, *light caching*) como motores secundarios. En este apartado lo utilizaremos como motor único para analizar mejor sus características.

La denominación *Deterministic Monte Carlo* (DMC) ) en la versión 1.7 de V-Ray y *Quasi Monte Carlo* (QMC) en versiones anteriores de V-Ray obedece a que se basa en una modificación de los métodos de Montecarlo por la que los puntos no se escogen aleatoriamente sino a partir de una función de distribución específica. El método utilizado está basado en una variante del propuesto por Christophe Schlick (ver Schlick, 1991). El método anterior, QMC, estaba basado en otra variante por la que las muestras se obtenían a partir de secuencias de números con propiedades especiales, mediante una técnica denominada “secuencias de baja discrepancia” (*low discrepancy sequence*). En el capítulo 4 ya he resumido las características principales de estos métodos por lo que remito al lector a las secciones correspondientes de ese capítulo.



Los resultados son también más precisos también por lo que respecta a la representación de los colores pues sus variaciones locales son más matizadas y responden con más fidelidad al modo en que estas variaciones se darían en la realidad. Pero esta mayor precisión puede quedar perturbada por el ruido remanente que hará que los colores resulten algo más oscuros por la sencilla razón de que los colores correctos, correspondientes a las muestras tomadas quedarán rodeados de colores incorrectos, más oscuros, no muestreados.

Otra ventaja digna de tenerse en cuenta es que la configuración es más sencilla pues hay menos parámetros que ajustar.

No es un método adaptativo, como *irradiance* o *light cache*: dispara por igual a zonas importantes que no importantes, de ahí el nombre de “fuerza bruta”: al no hacer valoraciones no puede equivocarse acerca de la importancia de una u otra zona con respecto a la imagen final. Esta es su principal ventaja y también su principal defecto como ya he dicho. También usa menos memoria pues no requiere, como los otros tres métodos, alma-

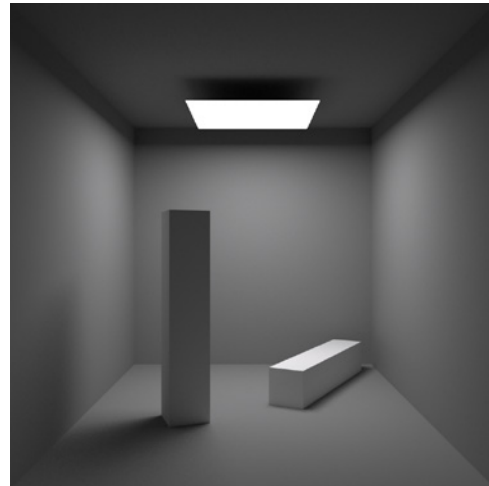
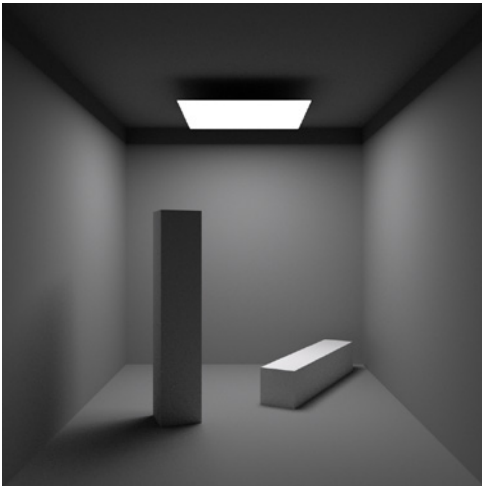


Figura 6.11 V-Ray. DMC (primario y secundario) con 3 bounces y diferentes valores de subdivisiones. Los tiempos se dan en relación a Light Cache (primario y secundario) con valores predeterminados (1X). Ver el texto para comparaciones más generales: a) 2 subdivs, tiempo 42X, b) 8 subdivs, tiempo 44X, c) 32 subdivs, tiempo 62X. Los resultados con 2 subdivs son prácticamente idénticos a los resultados con 8 y la diferencia de tiempo es pequeña. La variantes (a) y (b) corresponden a la figura de la izquierda

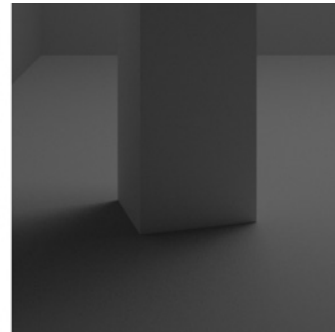
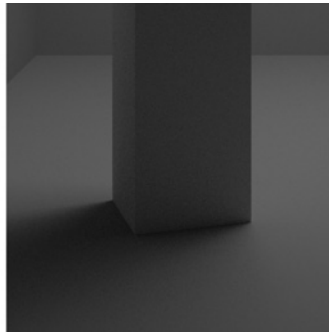
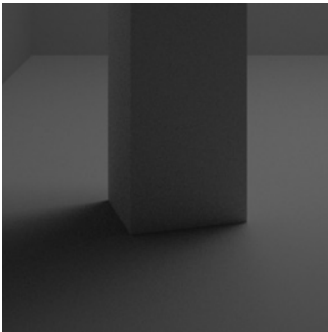


Figura 6.12 Figura 32 DMC. Detalle de la figura anterior: a) 2 subdivisiones, b) 8 subdivisiones, c) 32 subdivisiones



cenar un mapa de valores computados. Pero requiere mucho más tiempo para eliminar el ruido derivado de la ausencia de mecanismos de interpolación. Para alcanzar resultados de calidad suficiente puede ser necesario aumentar los valores predeterminados de 8 subdivisiones hasta 30 o más. Y los tiempos se dispararán en cualquier caso como puede comprobarse en los datos dados junto a las figuras adjuntas. Por esto solo es adecuado en escenas en las que se quiera mantener un alto nivel de detalle, tanto desde el punto de vista geométrico como del de las transiciones de color.

A diferencia de *Light Cache* que, en principio, calcula un número infinito de rebotes, DMC calcula un número limitado. Por esta razón los resultados pueden ser algo más luminosos con el primer método que con el segundo.

### Parámetros. Análisis

Los parámetros son los mismos tanto si se usa como primera opción o como segunda. Hay un único grupo, *Settings DMC*, con dos parámetros:

a) *Subdivs* (8). Determina el número de muestras que se toman para computar la Iluminación Global. El número de rayos trazados por V-Ray es proporcional al cuadrado de este número pero también depende de lo que se especifique en la sección *DMC Sampler* (*QMC Sampler* en versiones anteriores) que se detalla más abajo, en el apartado sobre *antialiasing*.

b) *Bounces* (3). Controla el número de rebotes de luz que se computarán.

Las figuras adjuntas muestran algunos ejemplos de variar estos parámetros y del tiempo de cómputo correspondiente utilizando DMC como motor primario y secundario. Tener en cuenta que en una escena de las características de la que estamos utilizando, modificar los rebotes no tiene consecuencias apreciables ni en la calidad de la representación ni en el tiempo: en las pruebas que he hecho los resultados son prácticamente idénticos

con 1 rebote (el mínimo) que con 3, que con 5, por lo que no mostro las variantes.

Por lo que respecta al **antialiasing**, la sección *DMC Sampler* (que substituye al *QMC Sampler* en versiones anteriores) incluye una serie de parámetros que permiten ajustar mejor el algoritmo. El número de muestras se determina por medio de tres factores: a) El valor de subdivisiones dado por la configuración de DMC que haya especificado el usuario (y cuyo valor predeterminado es 8 como ya hemos visto). Este valor puede cambiarse por medio de uno de los parámetros de esta sección, el multiplicador, "Subdivs multiplier"; b) La importancia del tipo de muestra, que requerirá una subdivisión mayor en unos casos, por ejemplo reflexiones brillantes, que en otros, por ejemplo reflexiones atenuadas, pues el efecto global, la repercusión de aquellas en el resultado final, será mayor. Esto es un ejemplo de aplicación de lo que, como ya hemos visto, se denomina genéricamente *importance sampling*; c) La variancia de las muestras tomadas para un punto particular que se traduce en más o menos ruido pues si los valores derivados de las muestras son muy homogéneos bastarán con pocas muestras. Y viceversa. Esto requiere una inspección de las muestras a medida que son captadas lo que puede activar un mecanismo de interrupción del muestreo (una forma de muestreo adaptativo denominada a veces "early termination").

Los parámetros con que nos encontramos y que están relacionados con estos tres factores, son los siguientes. Con *Adaptive Amount* (0.85) se controla hasta que punto se hace depender el muestreo de la importancia de los valores muestreado. El rango va de 1.0 (adaptación total) a 0.0 (adaptación nula). *Min Samples* (8) determina el mínimo número de muestras que deben tomarse antes de aplicar el algoritmo de terminación por suficiente homogeneidad de las muestras. *Noise Threshold* (0.01) establece el umbral a partir del cual se acepta que un valor es suficiente para ser utilizado, aplicándole cierto grado de des-



enfoque, aunque se introduzca cierta imprecisión en el resultado final. Cuanto menor sea el valor menos ruido se generará (0.0 implica que no se utilizará este mecanismo de adaptación al ruido). *Subdivs Mult* (1.0) es, como ya se ha indicado arriba, un multiplicador que afecta al número de subdivisiones dadas por el usuario lo que puede ser un modo rápido de reducir el tiempo para hacer pruebas (dando un valor inferior a 1.0) o, a la inversa, aumentar la calidad a costa del tiempo. Tener en cuenta que este parámetro puede afectar a cualquier combinación de algoritmos que utilice subdivisiones, no solo DMC sino también *irradiance* y los mapas de sombra generados por luces de área o reflexiones y refracciones. Pero no afecta a los mapas de fotones o de *light cache* que tienen su mecanismo propio de generación de muestras. *Path Sampler Type* ofrece tres opciones de generación del algoritmo y que son otras tantas variantes de métodos de *importance sampling*: *Schlick* (predet), *Legacy* y *Latin supercube*. Ver también el apartado sobre estos métodos en el capítulo 4 para una explicación básica de sus características.

## 6.4 Irradiancia y Final Gather

Como ya hemos visto, *Irradiance caching* calcula la irradiancia a partir de una muestra dispersa de puntos de la escena e interpola entre ellos durante la fase de *rendering*. Se crean nuevos puntos (*irradiance cache points*) sobre la marcha, enviando rayos sobre el hemisferio que rodea al punto si el error de interpolación es demasiado grande. Los puntos iniciales de cache se crean, “a menudo” desde cada píxel de una imagen remuestreada a la baja.

*Final Gather*, en general, es un método complementario, utilizado de diferentes modos, en combinación con radiosidad o fotones, que se da por cada píxel y que depende por tanto de la resolución final. Substituye valores erróneos por valores más exactos que se obtienen a partir de la evaluación, por métodos de Montecarlo, de la irradiancia de

puntos de la escena que son visibles desde el píxel. El método es tanto más exacto cuanto mayor es el número de rayos FG.

Por otro lado también hemos visto que, con *mental ray*, *Final Gather* se puede utilizar como un método independiente. En este caso hay una fase de precomputación y desde la cámara se envían rayos a la escena. En los puntos de intersección se guardan los puntos de la FG. Desde esos puntos se envían rayos en torno a un hemisferio en direcciones semialeatorias. Hay también un proceso de adaptación de modo que se acumulan más puntos en zonas críticas. Los rayos enviados reacumulan información sobre el color y la iluminación de otros puntos que se añaden al cálculo final. A esto le sigue una fase de computación adicional e interpolación. La fase previa es muy intensiva y depende del número de puntos especificados.

Aunque los métodos son muy similares, su implementación por programas concretos presenta diferencias importantes. En la primera parte de esta sección lo ilustraremos con la utilización de *Irradiancia* con *V-Ray* y en la segunda con la utilización de *Final Gather* con *mental ray*.

### Irradiancia con V-Ray

#### Descripción general

La escena está preparada tal como se ha descrito al comienzo de este capítulo, con algunas diferencias con respecto a otras pruebas, debido a las limitaciones del programa anfitrión que, en este caso, he elegido que sea *SketchUp*, para proporcionar una visión más amplia de las posibilidades y porque es una buena opción para quien no quiera utilizar programas de simulación excesivamente complejos. Como también he dicho, estas diferencias son poco importantes y, con la excepción del uso de planos de recorte que permiten ver el interior desde el exterior, todo lo demás, la geometría de la escena y las luces y cámaras utilizadas, son iguales.





Para configurar el cálculo, como también he dicho, todo lo que hay que hacer es activar *Options*, entrar en el panel *Indirect Illumination*, escoger los motores primario y secundario que queremos utilizar, seleccionando una de las cuatro opciones de cálculo, y ajustar los parámetros correspondientes de cada opción. Que en esta sección serán los de *Irradiance*.

### Parámetros generales

A diferencia de otras versiones de V-Ray que incluían un grupo de configuraciones de diferente nivel (*Built-in-presets* con configuraciones predeterminadas: *Very High*, *High*, *Medium*, *Low*, *Very Low* y *Custom*) la versión que he utilizado no las incluye. Como no cuesta mucho, es muy recomendable crearse pre-configuraciones propias. Todo lo que hay que hacer es presionar el botón de “guardar” que hay en la parte superior de *Options* y las configuraciones activas se guardan en un archivo en formato *visopt* que luego puede volver a cargarse para recuperar estas configuraciones.

Hay un gran número de **parámetros generales**. Los resumo brevemente, no con la intención de que el lector se zambulla en unas explicaciones que resultarán pesadas, sino para que cuente con una referencia adecuada a la que recurrir cuando sea necesario. En el siguiente apartado volveré sobre los principales.

El grupo *Basic parameters* incluye los siguientes. *Min/Max Rate* determinan la resolución inicial y final del cálculo y es uno de los parámetros principales que se comenta con detalle más abajo. Otro tanto cabe decir de *HSPH subdivs* que indica el número de rayos que se lanzan para calcular la iluminación en un punto a partir de una semiesfera que envuelve el punto y de *Samples*, que indica el número de muestras de interpolación que se tomarán y que también son parámetros principales sobre los que volveremos. *Clr threshold* (umbral de color, 0.3) afecta al contraste a partir del cual se toman más muestras (esto afecta principalmente a los bordes de

objetos y de sombras): si se disminuye el valor se pueden obtener más detalle en puntos críticos. *Nrm threshold* (umbral de normales, 0.3) computa la rapidez en la variación de la orientación de las normales. Si se disminuye el valor se tomarán más muestras en las zonas en la que una superficie se curve con rapidez. *Dist thr* (umbral de distancias, 0.1) hace que se tomen más o menos muestras en función de la distancia entre superficies. Un valor de 0.0 indica que no se tome en cuenta esta distancia. Valores superiores hacen que se tomen más muestras para objetos muy próximos. Puede ser conveniente subirlo (por ejemplo de 0.1 a 0.8) si hay esquinas o encuentros entre objetos claramente visibles, para que tome más muestras en esos puntos.

El grupo *Basic Options* incluye opciones para analizar el proceso. *Show calc. phase* (desact) puede ser útil para ver las muestras tomadas por el mapa de irradiancia. Los controles adyacentes, *Show dir. light* y *Show samples* sólo están disponibles si se activa el anterior y extienden estas visualizaciones que no afectan al cálculo.

El grupo *Detail Enhancement* se utiliza para añadir detalles adicionales en casos en los que haya detalles pequeños que pueden llegar a perderse. *Scale* se utiliza para elegir entre una escala basada en píxeles (*Screen*) o una escala basada en unidades de la escena (*World*). *Radius* determina el radio de este efecto: valores pequeños darán resultados más rápidos pero menos precisos pues las muestras se toman en áreas demasiado restringidas. Y viceversa. El significado es similar al del parámetro equivalente de *Ambient Occlusion*. *Subdivs mult* determina el número de muestras como un porcentaje del valor de *Hemispheric Subdivs*.

En el grupo *Advanced Options* se encuentran los siguientes parámetros adicionales: *Interpolation mode* se utiliza durante el proceso final de *render* y selecciona el método de interpolación de los valores de GI: *Least squares fit* (el valor predeterminado) busca el valor que se adapta mejor a las muestras, es más lento pero da mejores resultados;



*Weighted average* es más simple y rápido pero puede aumentar el ruido; *Delone triangulation* es mejor aunque tiende a emborronar los resultados; *Least squares with Voronoi* se supone que es una mejora de *Least squares*, aún más lento y con resultados no mucho mejores. Es mejor dejar el valor por defecto. *Sample lookup* se utiliza durante la fase de *render* y selecciona el método para escoger los puntos más adecuados del mapa de irradiancia como base para la interpolación: el método por defecto, *Density-based* es en general adecuado; *Nearest* es el más rápido (se limita a escoger los puntos más cercanos) y se utiliza para tanteos. Los otros métodos toman en cuenta otros factores, *Cal.pass interpolation sample* (15) se utiliza durante el cálculo del mapa de irradiancia, representa el número de muestras computadas que se usarán para guiar el algoritmo y los valores adecuados están entre 10 y 25; *Multipass* (act) se utiliza durante el cálculo del mapa de irradiancia para controlar si se toman en cuenta, o no, las muestras previas del cálculo, si se desactiva el resultado es algo más lento con imperceptibles mejoras en algunos casos por lo que lo más recomendable es dejarlo como está. *Randomize samples* (act) se utiliza durante el cálculo del mapa de irradiancia para que la toma de muestras relevantes en un determinado estadio sea aleatoria. Hay pocas razones para cambiarlo. *Check sample visibility* (desact) se utiliza durante la fase de *render*: si se activa quizás puede eliminar algunos artefactos que se empeñan en persistir pero es algo más lento pues se envían rayos adicionales para comprobar la visibilidad de las muestras.

El grupo *Mode* incluye una serie de opciones para especificar el modo en que se reutilizará el mapa de irradiancia. Da 6 opciones: *Single frame* (predeterminado) utiliza un único mapa para toda la imagen; *Bucket mode* usa un mapa distinto para cada region de render (*bucket*) lo que resulta útil en *render* distribuido pero no mucho en un único ordenador pues es ligeramente más lento; *Multi-frame incremental* es útil para escenas en las

que hay una secuencia de fotogramas en las que sólo se mueve la cámara; *From file* utiliza un mapa previamente almacenado y al que se accede con un botón anexo; *Incremental add to current map* utiliza un mapa previo que sólo se refina en donde se necesita. Se utiliza también para obtener múltiples vistas a partir de un mapa de irradiancia.

El grupo *Post Render* especifica que hacer con el mapa de irradiancia después de un *render*. *Don't delete* (predeterminado) significa que se guardará en memoria hasta el siguiente *render*. *Autosave* (desact) guarda automáticamente el mapa en el archivo especificado. *Switch to saved map* sólo está disponible con *Autosave* y, en este caso, se pasa de modo automático a *From file* (con el nombre del archivo recién guardado).

El grupo *Current Map* incluye dos botones *Save* (para guardar el mapa) y *Reset* para borrarlo.

## Análisis

Los **parámetros principales** que se utilizan en la práctica son *Min/Max*, *HSph Subdivisions* e *Interpolation Samples*. Las configuraciones mencionadas en cada caso se ilustran

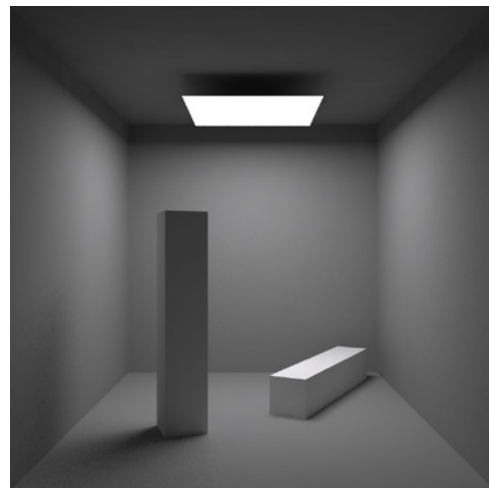


Figura 6.13 Irradiancia con V-Ray y valores predeterminados (Min,Max / HSph / Interpolation): -3,-1 / 50 / 20. Tiempo 1X

en las figuras adjuntas. La primera muestra los resultados al utilizar los valores predeterminados. Las siguientes, muestran las diferencias que aparecen al variar estos parámetros principales.

a) **Min/Max.** Un valor igual a 0 significa que se tomará una muestra por cada píxel, es decir que la resolución será igual a la de la imagen final. Un valor de -1, que se tomarán una muestra por cada 2 píxeles, es decir que la resolución será la mitad. Un valor de -2 que se tomará una muestra por cada 4 píxeles, es decir que la resolución será la mitad que la an-

terior y por tanto la cuarta parte de la original. Y así sucesivamente. Dicho de otro modo, si la resolución final es, por ejemplo,  $800 \times 600$ , un valor de *Min/Max*, -3/-1 (predeterminado) significa que se hará un primer cálculo (min) a una resolución de  $100 \times 75$ , un segundo cálculo, basado en el anterior, a  $200 \times 150$  y un tercer cálculo (max) a  $400 \times 300$ . La diferencia entre *Min* y *Max* está relacionada directamente con el número de pasos del cálculo de irradiancia. Así, si *Min/Max* está en -3/0 el cálculo se hará en 4 pasos. De ahí que, en las primeras pruebas, es conveniente que la diferencia entre estos valores sea la mínima

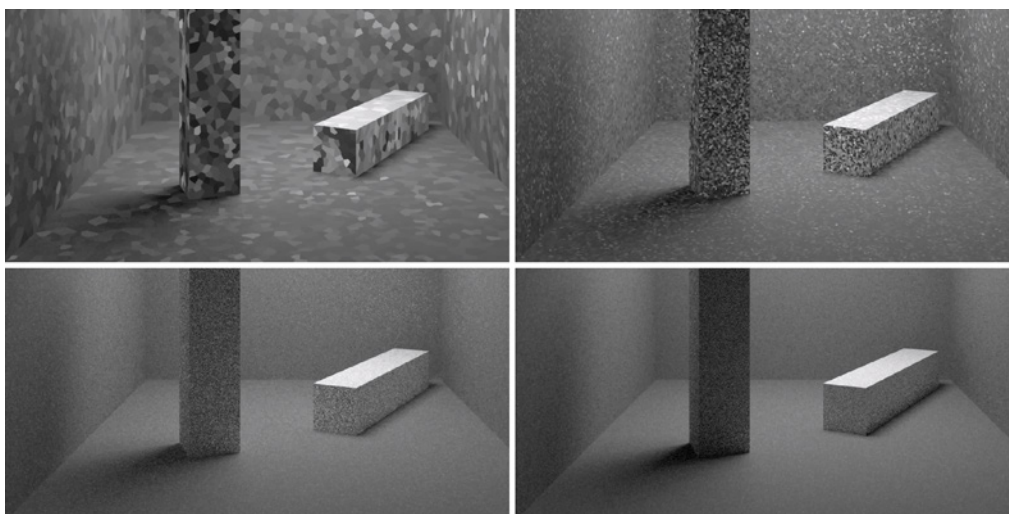


Figura 6.14 Irradiancia V-Ray, con HSph 1 e Interpolation Samples 1. Los tiempos se dan por referencia a los valores predeterminados (ver figura anterior): a) *Min,Max* -5, -3, tiempo 0.6X", b) *Min,Max* -3, -1, tiempo 1.8X, c) *Min,Max* -1, 1, tiempo 7.1X, d) *Min,Max* 1, 3, tiempo 16.2X

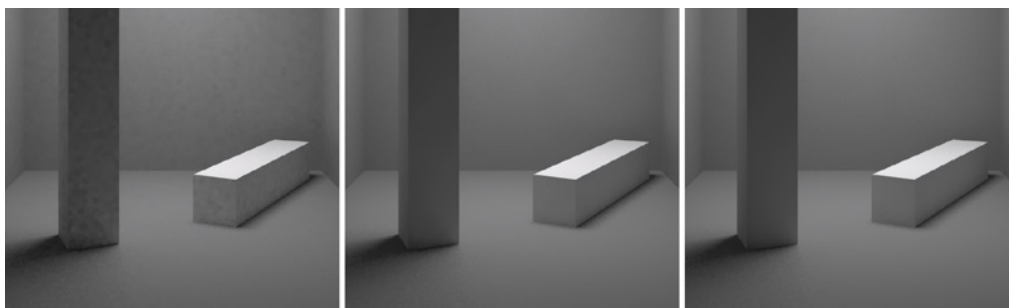


Figura 6.15 Irradiancia con V-Ray. Influencia de la interpolación. *Min. Max* -3, -1, HSph 20, Interpolación: a) 1, b) 20, c) 40. Los tiempos son similares



(por ejemplo -5,-4 o -3,-2).

b) **HSph subdivs** (subdivision hemiesférica, 50) indica el número de rayos que se lanzan para calcular la iluminación en un punto a partir de una semiesférica que envuelve el punto. Cuanto mayor sea el valor, mayor será la calidad general de la representación de una superficie (las superficies curvadas aparecerán más suaves). El número de rayos es proporcional (aproximadamente igual) al cuadrado del valor de *HSph* de modo que si se especifican 10 subdivs, el número de rayos estará en torno a 100, si 15, 225, si 20, 400, si 25, 625, si 50, 2500, etc. Esto se traduce en un aumento equivalente del tiempo de cálculo, es decir que si se aumenta este valor al doble, el tiempo de cálculo se multiplica por 4. El valor de *HSph* debe estar, en general, por encima de 20 y, en general, no muy por encima de 50 para que no se disparen los tiempos de cálculo.

c) **Interpolation Samples**. Este parámetro establece el número de muestras que se tomarán para interpolar los valores de iluminación indirecta. Valores altos darán resultados más suaves, más uniformes, a costa de la precisión pues pueden perderse pequeños detalles. Valores más pequeños darán resultados más precisos pero pueden dar lugar a manchas si el valor de *Hemispheric Subdivs* no es muy alto lo que puede ocasionar discontinuidades en el muestreo que se traducen en "manchas".

Esto quiere decir que tiene que haber un equilibrio entre estos dos últimos parámetros. Si el valor de *HSph* es bajo se requerirán más interpolaciones entre muestra y muestra. Y viceversa. Hay quien opina que es una buena regla hacer que ambos valores sumen 60 siempre que *Hsph* esté entre 20 y 40. Pero todo depende de los casos. *Hsph* no debe ser mucho mayor de 40 pues el tiempo aumenta exponencialmente. El valor de interpolaciones no se guarda con el mapa de irradiancia por lo que puede grabarse el mapa (ver más abajo) y modificar este valor antes de hacer un nuevo *render*.

Aunque no afecte demasiado al resultado final también debe tenerse en cuenta el valor de *Color Threshold* que puede revelar claras diferencias en el muestreo en el caso de imágenes en color.

Si se desactiva el segundo motor, podemos analizar los resultados debidos exclusivamente a irradiancia. La primera figura adjunta muestra los resultados que se obtienen con los valores predeterminados: -3-1 (min-max), 50/20/0.3 (subdivisiones, samples, umbral color).

Y reducir las subdivisiones y las muestras a 1, nos permitirá visualizar mejor en qué se basa el cálculo. Las siguientes figuras muestran los resultados característicos que se obtienen si solo se varía la resolución interna de la toma de muestras, los parámetros *Min*, *Max*, manteniendo en el mínimo los otros dos parámetros principales *HSph* e Interpolación.

Como he dicho, hay también otros parámetros que puede influir. Si, por ejemplo, se reduje el *Control Threshold* a 1.0 también se reduciría el número de muestras tomadas. Y en una escena con superficies curvas también tendría efecto reducir el *Control Normals*. La casuística es compleja pero la intención de esta sección es tan sólo ilustrar el sentido de los parámetros más relevantes.

## Final Gather con mental ray

### Procedimientos generales de configuración de iluminación global con *mental ray*

Cuando en 3ds Max se utiliza *mental ray* como motor de *render*, el aspecto de *Render Setup* muestra cinco secciones, dos iguales que las del motor de *render* predeterminado (*Common*, y *Render Elements*) y las otras tres, *Renderer*, *Indirect Illumination*, *Processing*, específicas de *mental ray*. La sección de *Renderer* incluye principalmente los controles de *antialiasing* que resumo en el siguiente párrafo. La sección *Processing* tiene tres subsecciones (*Translator options*, *Diagnostics* y *Distributed Rendering*) de la que, para los



propósitos de este capítulo, solo importa la segunda, *Diagnostics*, que permite visualizar las muestras que se toman de la escena al activar FG lo que facilita la comprensión del proceso. Pero no afecta en modo alguno al cálculo. La tercera sección, *Indirect Illumination*, incluye los parámetros generales de *Final Gather* que se describen más abajo.

El **antialiasing** se controla desde la sección *Renderer* del *Render Setup*. En esta sección hay cuatro subsecciones principales. La principal y la que nos importa en este contexto es la primera. Ver la sección sobre *aliasing* y *antialiasing* y filtros en el capítulo 4 para una explicación más genérica y completa de estos recursos. Aquí me limitaré a dar indicaciones prácticas.

En dicha sección, *Sampling Quality* se ajustan los parámetros de muestreo y antialiasing. El número de *Samples per Pixel* es el parámetro principal en el que se especifica el número de muestras que hay que tomar por cada píxel. Los valores *Min / Max* van de 1/64 a 1024. Lo habitual es comenzar por valores bajos para pruebas (por ejemplo 1/16 a 1/4) e ir aumentando progresivamente. Si se utiliza la ventana *Rendered Frame Window* para poner en marcha el proceso de *rendering*, esta ventana incluye configuraciones simplificadas que son perfectamente adecuadas para la mayoría de los casos: *Draft* (Min 1/64, Max 1/4), *Low* (Min 1/16, Max 1), *Medium* (Min 1/4, Max 4), *High* (Min 1, Max 16), *Very High* (Min 4, Max 64). En general, bastará con utilizar el primer ajuste, *Draft*, para hacer pruebas y *Medium* o *High* para las imágenes finales. Pero el valor adecuado está directamente relacionado con la resolución de la imagen y con su destino, si a va ser impresa sobre papel o proyectada en pantalla.

Los valores de umbral de *Spatial Contrast* afectan a que se tomen más o menos muestras cuando varían los colores de los objetos (por efecto de una luz rasante o un cambio de curvatura). Los valores predeterminados son 0.05 para los cuatro canales rgba. Las diferencias, sobre todo por debajo de los 0.1 son

poco perceptibles y en raras ocasiones merecerá la pena utilizar este ajuste.

Los tipos de *Filtro* son una selección de los principales que ya se encuentran en 3ds Max: *Box* (que utiliza un prisma como curva de distribución) es el más rápido pero también el que puede, en algunos casos, reducir el nivel de detalle. *Gauss* suaviza los resultados de un modo más ponderado, con una velocidad media. *Triangle* es algo mejor que *Box* y algo más lento. *Mitchell* (que utiliza una curva de distribución más eficaz) es más preciso pero más lento; puede hacer que zonas de poca resolución queden mejor definidas. Ver la sección correspondiente del capítulo 4 en donde se amplía la información sobre estos filtros. Pero tener en cuenta sin embargo que las diferencias también resultarán poco apreciables en la mayoría de los casos.

Las *Opciones* se utilizan principalmente para animación (*lock samples* o *jitter*) y para cambiar el tamaño de las "bolsas" (*buckets*) que se procesan durante el proceso de *rendering*.

A partir de la versión 2014 de 3ds Max hay recursos adicionales de *antialiasing*, un control global de *Sampling Quality*, a los que también me he referido en el capítulo 4 por lo que me remito a ese capítulo. Es un recurso sencillo de utilizar pues prácticamente basta con especificar un determinado nivel de calidad, con valores bajos para pruebas y altos para salidas finales. El rango es de 0.1 a 20.0 con un valor predeterminado de 0.25.

### Descripción general

El procedimiento básico de *Final Gather* como procedimiento independiente de cálculo ya se ha descrito con carácter general en el capítulo 3. Aquí lo resumo muy brevemente para relacionarlo con los parámetros concretos de *mental ray* que utilizaremos.

a) Desde la cámara se envían rayos hacia la escena. Estos rayos se encuentran con las superficies de la escena en diferentes puntos escogidos semialeatoriamente. Cuando el rayo encuentra un punto (denomina-





do *FG point*) se computan sus propiedades materiales, su *material shader* (si el rayo no encuentra superficies se computa el *environment shader*). En cualquier caso se hace una media de todos esos rayos para encontrar la radiancia media. Digo que son escogidos “semialeatoriamente” pues la distribución inicial de los rayos con FG depende generalmente de una estimación global de la geometría, el contraste de iluminación, etc, es decir, de alguna variante de *importance sampling*. Si se activan los diagnósticos (en *Render Setup/ Processing/ Diagnostics*) con diferentes configuraciones de FG es notorio que la distribución varía y tiende a acumularse en zonas críticas (esquinas, cambios de gradiente, etc).

b) Desde cada uno de los *FG points*, se envían nuevos rayos, también de modo semialeatorio hacia la escena. Estos rayos registran la iluminación indirecta correspondiente a los puntos de las superficies con que se encuentran. Los rayos se distribuirían sobre el hemis-

ferio que rodea el punto utilizando una función ponderada. Pero con FG no se envían estos rayos sobre el hemisferio cada vez que se encuentra un punto y se computa su *material shader*. En lugar de eso, se calculan todos los *FG points* de la escena y se utilizan los puntos más cercanos para interpolar un valor cuando se computa el *material shader*.

c) Si el número de rebotes (*bounces*) es mayor que 0 se repite el proceso tantas veces como quede determinado por el número de rebotes y desde cada nuevo punto de intersección se envían nuevos rayos.

d) A medida que se procede de este modo, parte de la intensidad se atenúa con la distancia (FG sigue la ley de atenuación cuadrática inversa) y parte se atenúa debido a que es absorbida por las superficies en mayor o menor grado, y su color se modifica debido a que esta absorción es mayor en unas longitudes de onda que en otras. La intensidad y color finales se almacenan en el último punto al que

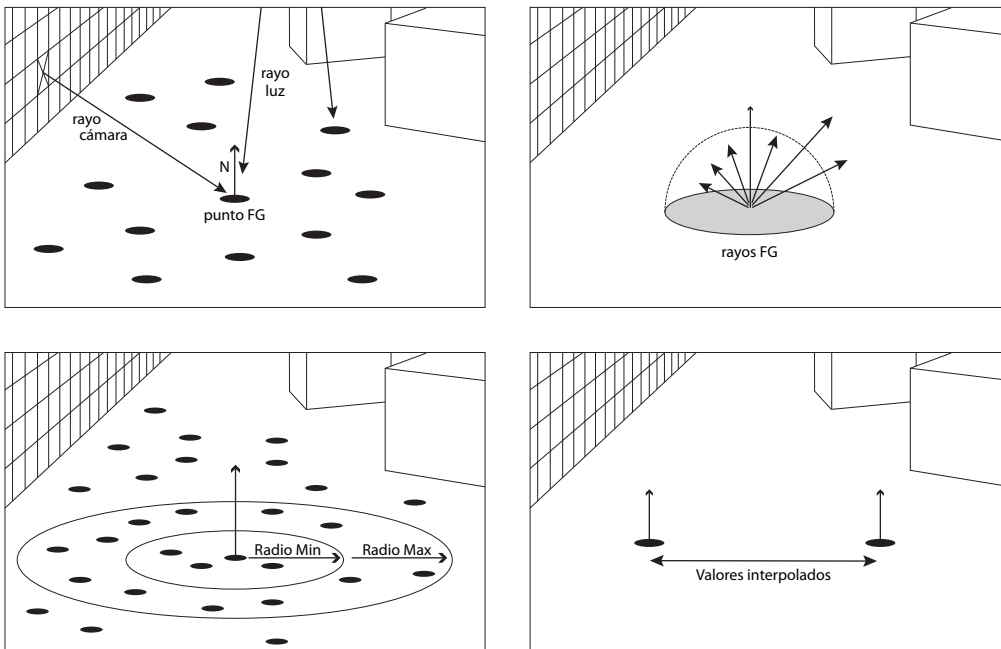


Figura 6.16 Final Gather. Esquema de los parámetros principales: a) Puntos FG distribuidos según densidad, b) rayos enviados desde cada punto FG hacia su entorno inmediato, c) puntos FG distribuidos según radios mínimos y máximos, d) valores interpolados entre dos puntos FG





llega el rayo. Todos estos puntos se almacenan en un mapa 3D, como una nube de datos ligada a las superficies y que se puede visualizar con las herramientas de diagnóstico.

Hasta aquí el cálculo de FG. El cuadro de diálogo que se abre al comenzar el proceso informará que la tarea actual es "Computing Final Gather Points". Al activarse el proceso de render, el cuadro de diálogo informará que la tarea actual es "Rendering".

e) Durante el proceso de *render*, la iluminación guardada en los diferentes puntos se pondera para interpolarla en un radio especificado por los parámetros de cálculo y en una cantidad también especificada por estos parámetros. Y se añade a la información sobre el color correspondiente a cada superficie, que dependía inicialmente del color del material y de su modificación por la luz directa.

En la fase de *render* se cuenta por tanto con una serie de *FG points* que almacenan información (color e intensidad) que se ha obtenido por medio de este cálculo y que ya incorpora una interpolación. Se dice que el precalculo está cogiendo (*caching*) resultados que corresponden a la iluminación indirecta difusa. Estos resultados se combinan por los obtenidos por otras vías.

Hasta la versión 2009 de 3ds Max, FG se calculaba en un único pase de *render*. En las versiones posteriores se calcula en tres pases. Es conveniente observar los resultados en el primero o segundo pase y cancelar el render si se aprecia que estos resultados no son los que interesan.

La primera figura adjunta ilustra este proceso y los parámetros generales implicados.

### Parámetros

A diferencia de Irradiancia con V-Ray, con FG hay muy pocos parámetros y bastará en la gran mayoría de los casos con ajustar los tres principales que se resumen en los siguientes apartados.

La configuración del cálculo se establece por medio de los parámetros generales en la sección *Indirect Illumination* que se discute a continuación. Esta sección tiene dos subsecciones principales. La primera es la principal y la única utilizada en la mayoría de los casos. La segunda permite activar fotones y cáusticas y se explicará en el subcapítulo dedicado a Fotones.

En esta sección principal, *Final Gather*, encontramos los siguientes grupos:

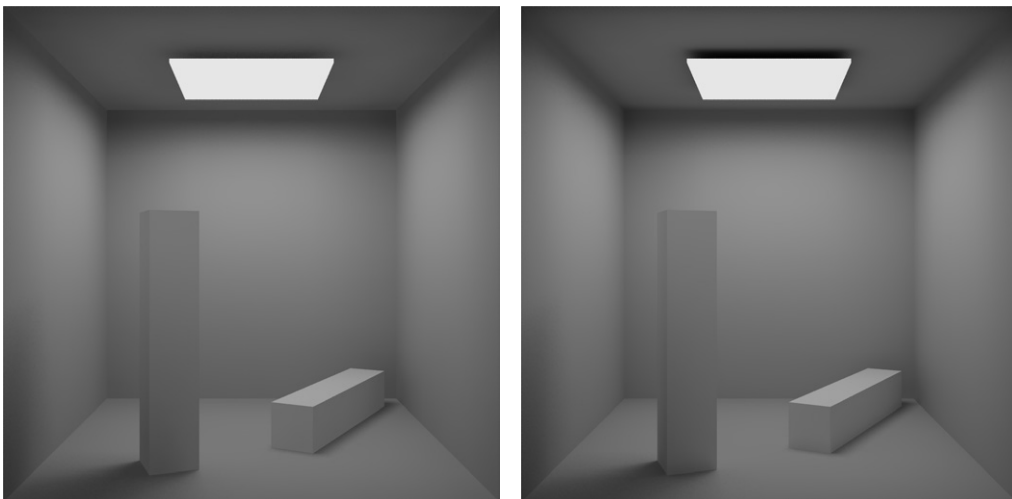


Figura 6.17 Final Gather. Resultados que se obtienen con las configuraciones predeterminadas. Los valores indicados entre paréntesis corresponden a Densidad/Rayos e Interpolación: a) Draft (0.1/50/30). Tiempo 1X, b) High (1.5/500/30). Tiempo 23X

El grupo *Basic* contiene una serie de parámetros que controlan la precisión del cálculo y que pueden ajustarse automáticamente mediante los *Presets* que figuran en la parte superior y que proporcionan 5 niveles, de menos a más (“Draft”, “Low”, “Medium”, “High”, “Very High”). Si se escoge *Custom* se supone que los valores serán personalizados. Si

se escoge cualquiera de los *Presets* pero se cambia algún valor, el tipo de configuración pasa automáticamente a *Custom*. El multiplicador y el icono de color permiten modificar la intensidad y el color de la iluminación indirecta global.

En la mayoría de los casos bastará con escoger alguna de estas configuraciones pre-



Figura 6.18 Final Gather. Variaciones de resultados con Densidad 0.1 y con: a) Rayos 1 (tiempo 2.5X), b) Rayos 10 (tiempo 1.2X), c) Rayos 100 (tiempo 1X)

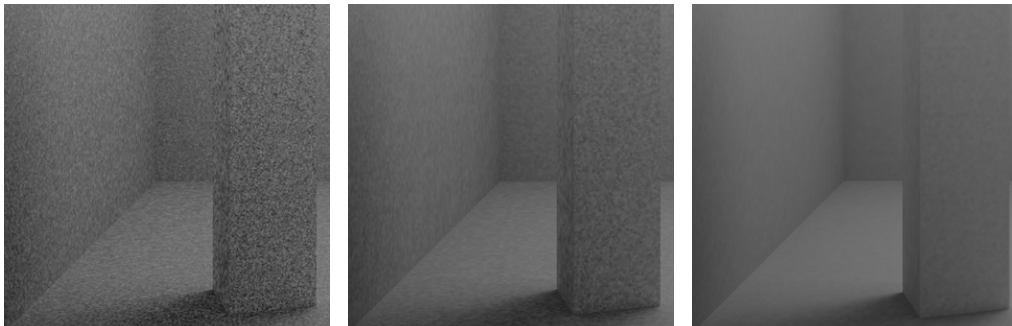


Figura 6.19 Final Gather. Variaciones de resultados con Densidad 1.0 y con: a) Rayos 1 (tiempo 5X), b) Rayos 10 (tiempo 2.3X), c) Rayos 100 (tiempo 2.5X)

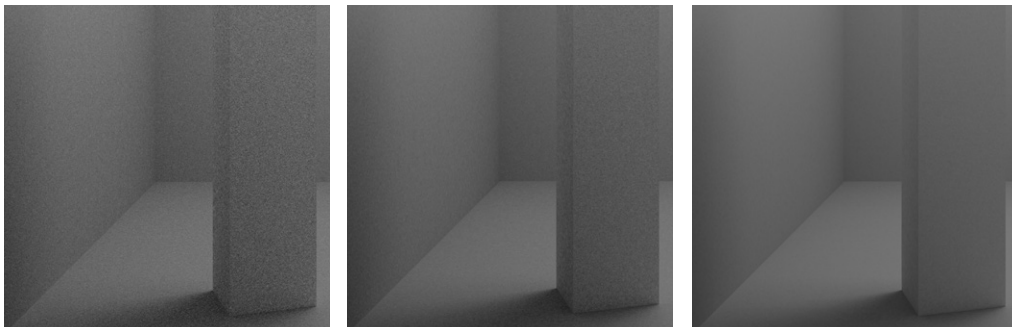


Figura 6.20 Final Gather. Variaciones de resultados con Densidad 10 y con: a) Rayos 1 (tiempo 10X), b) Rayos 10 (tiempo 7.5X), c) Rayos 100 (tiempo 15X)



determinadas. Las dos figuras muestran los resultados que se obtienen al utilizar los dos extremos, “Draft” y “High” (descarto “Very High” pues las diferencias son inapreciables en la gran mayoría de los casos y los tiempos se disparan).

Como se puede apreciar en estas figuras, las diferencias, para una escena simple como la que estamos utilizando, son poco relevantes.

Pero precisamente porque hay muchos casos que requieren ajustes más precisos conviene conocer el sentido de los tres **parámetros principales** que afectan a la calidad del resultado:

a) **Densidad de puntos FG.** Este parámetro, *Initial FG Point Density* (1.0) permite controlar la densidad de los puntos iniciales. Si se

aumenta este valor, aumenta el número de puntos que se envían a la escena desde la pantalla de salida. Estos puntos se distribuyen de un modo regular (alrededor de uno por cada 10 píxeles para una densidad de 1.0) y dependen de la resolución. Hay que aumentar la densidad si se necesita más detalle y reducirla para pruebas previas.

b) **Número de rayos FG.** Este parámetro, *Rays per FG point* (250) determina el número de rayos que se envían por cada muestra para computar la iluminación indirecta correspondiente al punto muestreado (este valor se ha reducido a partir de la versión 9.0 de 3ds Max). Hay que aumentar el número de rayos si la escena tiene un contraste alto y necesitamos que los gradientes de intensidad estén bien representados. Un valor en

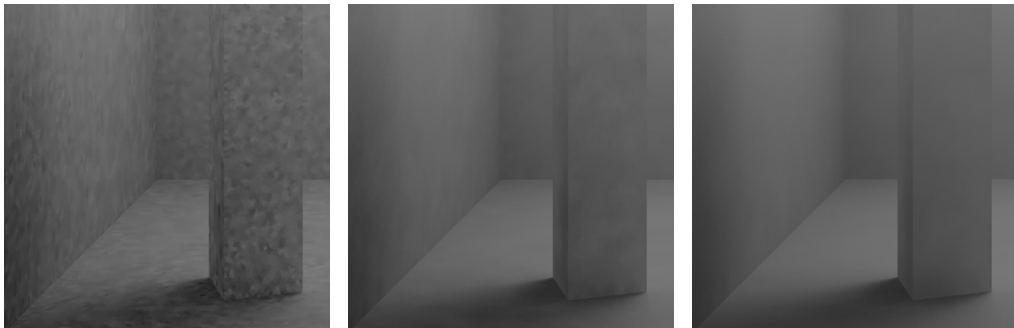


Figura 6.21 Final Gather. Influencia de la interpolación (Densidad 0.1, Rayos 10): a) Interpolación 1 (tiempo 1.1X), b) Interpolación 10 (tiempo 0.9X), c) Interpolación 30 (tiempo 0.9X)

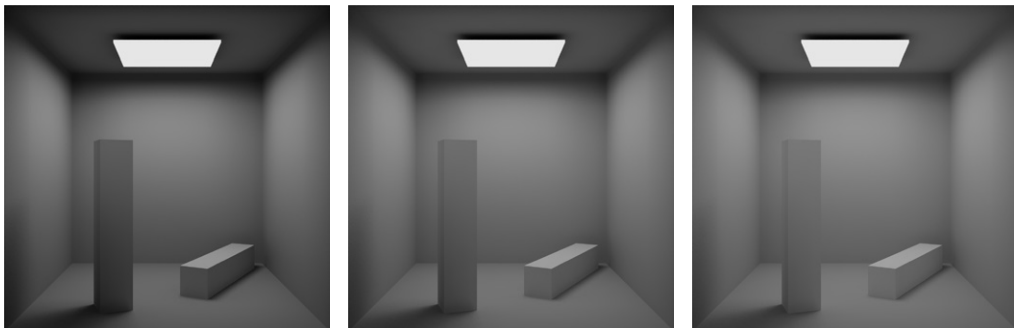


Figura 6.22 Final Gather. Influencia del número de rebotes: a) 0 bounces (tiempo 1X), b) 1 bounce (tiempo 1.7X), c) 2 bounces (tiempo 3X)



torno a 100 puede ser suficiente para escenas de bajo contraste. Pero una escena de alto contraste puede requerir 1.000 o incluso 10.000.

Hay que calibrarlo bien pues si el ruido residual no afecta al detalle es preferible bajar el número de rayos y aumentar un poco la interpolación.

El detalle y la calidad pueden aumentar de ambos modos. En escenas con alto contraste (iluminaciones intensas en una zona y zonas oscuras en otras) resultará necesario aumentar el número de rayos secundarios enviados a la escena desde los puntos FG (1.000 o hasta 10.000). Por el contrario, en escenas con bajo contraste, iluminadas de modo regular, se necesitarán menos rayos (50 o menos). En los casos generales un número comprendido entre 100 y 500 será suficiente.

En las figuras adjuntas se analizan, en una vista de detalle, las diferencias de calidad y tiempo que se obtienen al variar la densidad y el número de rayos. El tiempo se ha computado en relación con los valores más bajos, correspondientes a un tiempo de unos 12" (1X) en un ordenador de calidad media. El valor de interpolación se ha fijado en todos los casos en 1, la calidad de *antialiasing* en valores medios (min /1/4, max 4) y el número de rebotes en 2.

c) **Interpolación.** Este tercer parámetro, *Interpolate Over Num FG Points* especifica el máximo número de puntos que se tomarán en consideración de los que rodean al que se calcula. Es un parámetro que, en versiones más recientes, substituye a los parámetros de "radio máximo" y "radio mínimo" de versiones anteriores, en las que el valor 1.0 como "Radio máximo" abarcaba toda la escena y la reducción de este valor aumentaba la calidad. El valor de "Radio mínimo" sólo estaba disponible si se activaba el anterior y su valor por defecto era de un 10% del máximo.

Tener en cuenta que si se da un valor muy alto de interpolación todos los puntos

se suavizan con lo que se eliminan muchos defectos. Pero se pierde detalle. Y la pérdida de detalle puede ser importante en una escena que incluye líneas finas, zonas de alta frecuencia.

Para recapitular: a) Densidad implica número de muestras (densidad de muestras) que se toman desde la cámara, b) Rayos por FG implica calidad de muestras: si cada muestra envía muchos rayos de inspección a su alrededor sabrá mejor que es lo que le llega y cuáles son los valores que debe tomar para aproximarse a la realidad (virtual), c) Interpolación implica suavizar los defectos a costa de un cierto riesgo de comerse detalles importantes.

Aumentar la densidad tiene escaso efecto a partir de cierto límite. Con una configuración de densidad 100, rayos 100 y sin interpolación, los resultados serán más precisos (de hecho es equivalente a utilizar *path tracing*) pero el tiempo se dispara hasta cerca de 180X con respecto a una densidad de 0.1 y 100 rayos y hasta cerca de 12X con respecto a una densidad de 10 y 100 rayos sin que haya una gran diferencia con respecto a este último resultado que se ilustra en las figuras adjuntas.

Además de estos tres parámetros principales hay un cuarto parámetro que también afecta de un modo significativo al resultado: el número de rebotes que se computan. Con *Diffuse Bounces* se especifica este valor, el número de rebotes (0 por defecto) que se computan para cada rayo. El parámetro adicional *Weight* (1.0) controla la contribución relativa de los rebotes a la solución final. Puede merecer la pena aumentar este valor en escenas con rincones que quedan demasiado oscuros.

En la figura adjunta se muestra las diferencias de utilizar 0, 1 o 2 rebotes. Si la escena requiere más rebotes es preferible combinar FG con fotones, como veremos más adelante, que aumentar el valor de este parámetro.



## 6.5 Fotones. Combinación de irradiancia y fotones. Cáusticas

Como ya hemos visto en el capítulo 3, los mapas de fotones almacenan los fotones en una estructura de datos tridimensional especial, por lo general un *Kd-tree*, una técnica de partición espacial sistemática que facilita la inspección rápida de sus elementos y de sus vecinos cercanos, de los elementos que caen dentro de un determinado rango. Esta facilidad de recuperación rápida de valores en un entorno (que en el caso de los fotones vendrá dado por el radio asociado a los fotones) es la razón por la que se utiliza este tipo particular de estructura.

También hemos visto que el cálculo de Iluminación Global por medio de fotones se basa en un proceso en dos fases. En la primera fase se almacena la distribución de fotones en la escena, grabando sus valores de energía en formato RGB. Estos valores sólo se registran si la superficie que el fotón encuentra durante el proceso, cuenta con especificaciones de reflexión adecuadas y refleja luz difusa. Si la superficie fuera completamente negra o completamente especular o completamente transparente no se registraría ningún valor: el fotón se absorbería, se reflejaría, o se refractaría sin dejar rastro. Si hay un valor registrable, este se almacena en el árbol *Kd*. Es decir que los fotones llevan consigo una fracción de la intensidad de la luz que los emite. Una luz más intensa emite más fotones que otra menos intensa. Los fotones se reflejan en las superficies difusas transportando energía y color de superficie a superficie. En cada rebote, su valor se modifica en función de las características de la superficie que absorberá parte de su energía y color, excepto en el caso de superficies perfectamente reflectantes o transparentes en las que el fotón no se modifica. En la primera superficie de contacto no se almacenan fotones porque esto se computa como iluminación directa. De este primer contacto surgen nuevos fotones con diferentes direcciones, más o menos aleatorias, en función del tipo

de reflectancia de la superficie y con energía y color modificados. Habrá más fotones en direcciones especulares que en otras y muchos menos cuanto más se aleje el ángulo de la dirección especular. Estos nuevos fotones llegan a otras superficies hasta que el proceso termina.

En la segunda fase, se calculan los valores de la energía que llega a un determinado punto, promediando los valores del entorno de ese punto a partir de un determinado radio que es uno de los parámetros característicos de configuración del cálculo por fotones y que permite tomar los valores de los fotones almacenados en el árbol *Kd* que caen dentro de este radio. Estos valores así promediados son los que pasan al motor de *render* para que derive de ellos los valores de cada píxel.

En la primera fase se graba por tanto el mapa de fotones (que, como veremos más adelante, se puede grabar para reutilizarlo saltándose así la primera fase en sucesivos recálculos). Durante la segunda fase (de *render*) todavía se puede ajustar el modo en que se interpolan los fotones, qué radio se aplicará y cuantos fotones intervendrán en este promedio. De hecho la interpolación se inicia durante la fase de toma de muestras para iniciar el *rendering*. Pues cada muestra computa los valores a partir de la información sobre los elementos cercanos que le es suministrada por este cómputo previo. Y de este modo es como se calcula el valor de la iluminación indirecta que llega a ese punto. Este es el dato que finalmente se pasa al motor de *render* para que aplique la ecuación de *render* combinando las contribuciones de iluminación directa e indirecta.

El cálculo de fotones, a diferencia de FG, no depende del punto de vista sino de la posición de las luces y por tanto es válido para diferentes puntos de vista. El mapa de fotones puede guardarse y puede utilizarse para diferentes vistas mientras no se hayan modificado las posiciones de los objetos o las luces. El trazado de fotones es un caso de lo que se conoce como *forward ray tracing* y que sigue un camino inverso del *ray tracing* clásico que



envía rayos desde la cámara hacia la escena. En el caso de los fotones o, en general, de *forward ray tracing*, los fotones se originan desde la luz. Y, a diferencia de otros sistemas similares, los fotones almacenan energía (radiancia) y los rayos se utilizan para transportar los fotones y para reflejarlos o refractarlos en función de las propiedades de la superficie.

Todo este proceso no varía significativamente según las diferentes aplicaciones aunque si puede variar la terminología y la interfaz.

## Fotones con mental ray

### Descripción general

La descripción dada en el apartado anterior resume adecuadamente el proceso que se sigue con *mental ray*. Y también con *V-Ray*. No hay diferencias significativas pero describiré el procedimiento con ambos sistemas.

Puede convenir excluir objetos tales como vidrio o espejos del cálculo de fotones. Para ello, seleccionarlos y en *Objects Properties/mental ray*, marcar la opción *Exclude from GI Calculations*.

### Parámetros generales

Hay pocos parámetros importantes que ajustar. En este apartado describo brevemente todos y, en el siguiente, los tres principales.

La configuración del cálculo de fotones se ajusta, como FG, desde *Indirect Illumination* que tiene dos subsecciones principales. En el caso de los fotones los ajustes se hacen desde la segunda. En esta segunda sección principal, *Caustics and Global Illumination*, encontramos los siguientes grupos.

El grupo *Global Illumination* contiene un multiplicador y un modificador de color similares a los de FG. Los dos primeros parámetros, *Maxim Num Photons per sample* y *Maximum Sampling Radius* se describen con más detalle más abajo pues son dos de los tres principales. Con *Merge Nearby Photons* se puede

optimizar el uso de la memoria, haciendo que los fotones que estén a una distancia inferior a la dada por el valor de umbral se fusionen (el valor por defecto, 0.0 equivale a desactivar esta opción). *Optimize for FG (slower GI)* almacena información adicional para cada fotón (valores de iluminación de sus vecinos) lo que hace el cálculo de IG más lento pero, si se utiliza IG en combinación con FG, reduce el tiempo total.

El grupo *Trace Depth* incluye dos parámetros que controlan el número máximo de reflexiones y refracciones y el número máximo de la combinación de ambas (si la suma es 5 puede haber, por ejemplo, 3 reflexiones y 2 refracciones, pero no 3 y 3).

El grupo *Light Properties* afecta a todas las luces e incluye un parámetro principal, *Average GI Photons per Light* que también se describe con más detalle a continuación. El valor de *Decay*, especifica como disminuye la energía fotónica con la distancia. Puede ser ninguna (0.0), inversa (1.0) o cuadrática (2.0) que es la predeterminada. O valores intermedios.

El último grupo, *Geometry Properties*, incluye una única opción para hacer que todos los objetos reciban y generen fotones de iluminación global (GI) y cáusticas. En general, sobre todo en el caso de cáusticas, como veremos, es preferible hacerlo manualmente para no sobrecargar el cálculo.

### Análisis

Los tres parámetros principales se describen a continuación y sus variaciones se ilustran en las figuras adjuntas.

a) **Número de fotones** (*Average GI Photons per Light*). Desde aquí se especifica el número total de fotones que se envían por término medio desde cada luz. Como veremos en el ejemplo que sigue, es preferible ir tanteando valores, comenzando por valores del orden de 10.000 a 100.000 pues en ocasiones puede ser necesario llegar a más de 1 millón lo que alarga el tiempo de cálculo y conviene no utilizar más de los necesarios para no alargar este tiempo.





b) **Radio de los fotones Maximum Sampling Radius** (desactivado) permite especificar un valor para el radio de los fotones. Si no se especifica, se toma un valor que suele ser del orden del 10% de la escena. Si se especifica, el valor predeterminado es 1.0 y habrá que variarlo en función de las características geométricas de la escena o de lo que nos interese, tomando como referencia inicial este 10% de la dimensión mayor de la escena que estemos simulando.

c) **Número de fotones por muestra** (*Maxim Num Photons per sample*) determina el número de fotones máximo que se enviarán para cada muestra. Como siempre, será conveniente comenzar por valores bajos, por debajo de 100, e irlos aumentando. Pero tener en cuenta que aumentar este valor suaviza el resultado a costa de disminuir la precisión de los detalles. Si se quiere obtener resultados precisos habría que aumentar el número de

fotones, no el número de muestras interpoladas. El valor predeterminado es 500.

Para entender adecuadamente el número y tamaño de fotones necesarios y el modo en que contribuyen a calcular la iluminación global de una escena lo mejor es comenzar por valores pequeños e irlos aumentando. Las figuras que se adjuntan se han obtenido con las siguientes configuraciones.

1. Ir a *Render Setup/Indirect Illumination/Final Gather* y desactivar FG.
2. Ir a *Render Setup/Indirect Illumination/Caustics and Global Illumination* y activar *Global Illumination*.
3. Cambiar los siguientes valores:  
En *Light Properties/Average GI Photons...* dar un valor inicial de 10.000.  
En *Maximum Sampling Radius*, dar un valor inicial de 1 para ver con claridad los fotones.

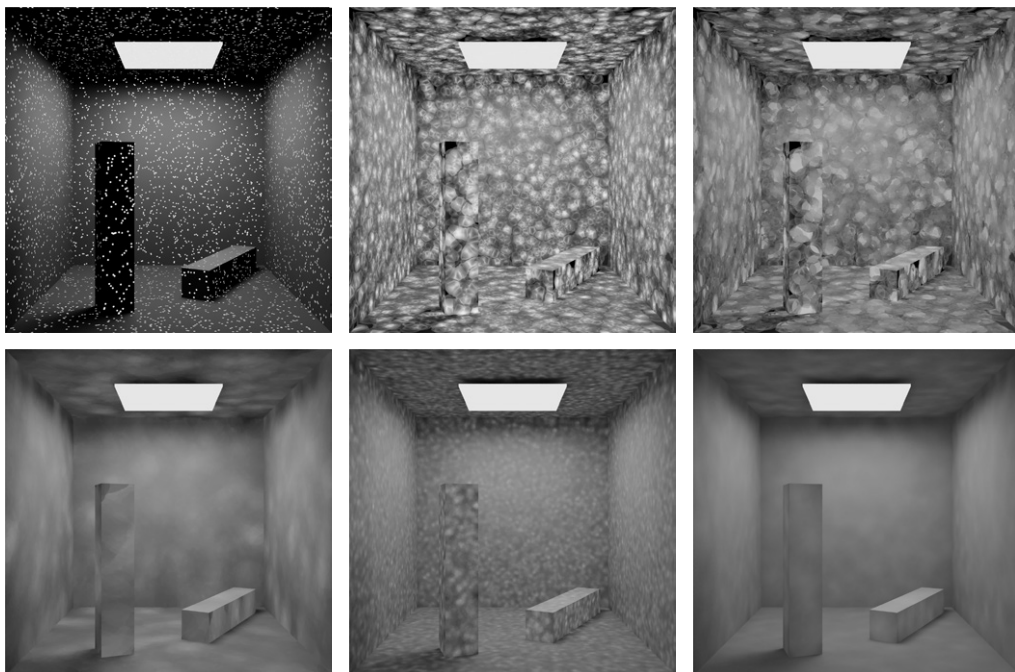


Figura 6.23 Fotones con mental ray. Configuraciones de cálculo (número de muestras / radio / número de fotones, tiempo): a) 1 / 1.0 / 10.000 (tiempo 1.5X), b) 1 / 10.0 / 10.000 (5X), c) 10 / 10.0 / 10.000 (2.5X), d) 30 / 30 / 10.000 (1X), e) 30 / 30 / 300.000 (2X), f) 300 / 30 / 300.000 (1.2X)

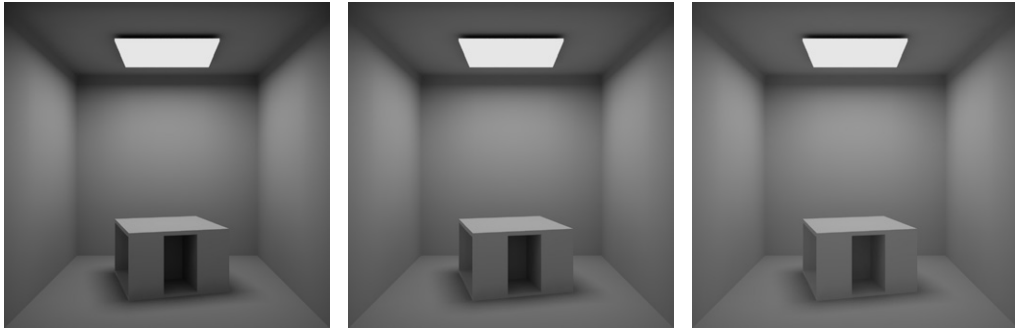


Figura 6.24 Solo FG con calidad media y tiempos en relación con solo fotones: a) 1 rebote, tiempo 3.7X, b) 2 rebotes, tiempo 4.4X, c) 4 rebotes, tiempo 5.1X

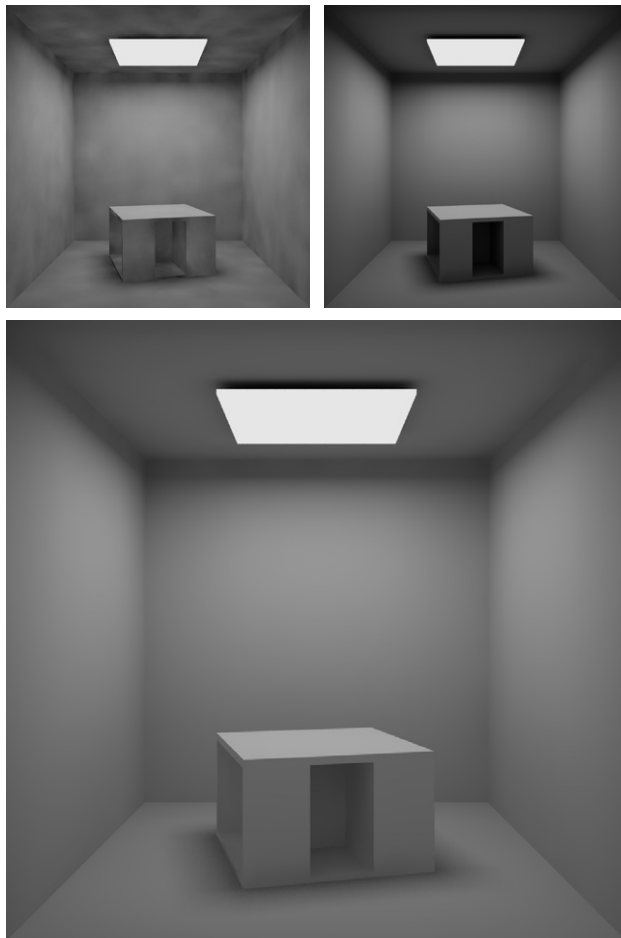


Figura 6.25 a) Fotones sin FG, tiempo 1X, b) Solo FG, 0 rebotes, tiempo 2.5X, c) Fotones y FG con 0 rebotes, tiempo 4X



En *Maximum Number of Photons per sample*, dar un valor inicial de 1 por la misma razón.

4. Activar *Render* para comprobar el resultado que deberá ser como el de la primera figura del grupo de seis figuras adjuntas. Ir aumentando el valor del radio hasta que los fotones se solapen lo que deberá corresponder más o menos a un 10% de la dimensión mayor de la escena. Luego aumentar el número de fotones por muestra para que este solape sea más efectivo. Y luego ir aumentando el número de fotones para que la precisión del cálculo sea mayor. Los resultados deberán ser similares a los de las figuras adjuntas.

Comprobar los tiempos: valores pequeños no dan necesariamente resultados más rápidos. Hay valores óptimos, como los correspondientes a la figura (d) del grupo adjunto que indican que tenemos suficiente para pasar a combinar este resultado con FG como haremos más adelante. Y aunque aumentemos los valores el ruido no disminuirá. Enviar 3 millones de fotones no mejorará demasiado el resultado aunque los tiempos tampoco se incrementarán tanto como se podría pensar: lo que caracteriza al cálculo de fotones es que es muy rápido.

Para comparar los resultados y los tiempos, desactivar fotones y hacer 3 *renders* con las configuraciones indicadas en las figuras adjuntas (ningún rebote, 2 rebotes, 4 rebotes).

Los ajustes generales de calidad y ***antialiasing*** se hacen igual que con FG (ver el apartado correspondiente).

Las áreas alrededor de las luces de una escena pueden aparecer quemadas debido a un exceso de intensidad, principalmente cuando se usan fotones como apoyo al cálculo de FG en luces situadas dentro de luminarias. Si ocurriera esto probar alguna de estas correcciones:

- a) Mover las luces fotométricas alejándolas un poco de la luminaria.
- b) Cambiar la distribución de la luz (a *uniform diffuse* o *spot*) para dirigir los fotones lejos de la luminaria.
- c) Cambiar las propiedades del interior de la luminaria para que no reciba fotones.

### Combinación de irradiancia y fotones con mental ray

Como ya he comentado, el cálculo por fotones es más rápido que muchos métodos pero se necesita un gran número de fotones para llegar a un resultado que no muestre imperfecciones en las zonas de transición. Por esta razón requiere ser complementado con métodos de interpolación que suavicen los resultados, métodos que son propios del cálculo por irradiancia. Como estos tipos de cálculo han mejorado espectacularmente en los últimos años y la idea básica no es demasiado distinta, en la gran mayoría de los casos no será necesario recurrir al cálculo por medio de fotones.

Sin embargo hay dos excepciones: las escenas con rincones a los que no llega la luz fácilmente y las escenas en las que se producen reflejos y sombras cáusticas. En el primer caso es conveniente utilizar fotones porque, en combinación con irradiancia, produce mejores resultados y en menos tiempo. En el segundo caso hay que utilizar fotones pues de otro modo no se conseguiría la precisión requerida por este tipo de efectos.

La figura adjunta muestra un ejemplo sencillo que servirá para comprobar lo primero. Los prismas de la escena original se han substituido por un objeto que incluye rincones a los que no llega fácilmente la luz.

Como puede comprobarse en estas figuras, los resultados, tanto desde el punto de vista de la calidad o de objetivos específicos (los rincones quedan mejor iluminados) como desde el tiempo de cálculo, son más efectivos si se utiliza una combinación de fotones y *Final Gather*.



### Fotones con *mr sun*

Uno de los parámetros de *mr Sun* es *Photon target*. Esto se utiliza cuando queremos combinar fotones con FG en el caso de una luz diurna que ilumina un interior. No es una opción corriente pero, en el caso de que se utilice, merece la pena utilizar este parámetro pues si no se activa esta opción, el sol emitirá fotones en todas direcciones lo que supone un derroche inútil. Activando esta opción se restringe el área en que se emiten fotones, lo que resulta más efectivo. La única diferencia está en el tiempo de cálculo pero como puede comprobarse si llega el caso, la reducción de tiempo será significativa.

Al activar esta opción aparecerá un cilindro en torno a la línea que conecta el sol con la rosa de los vientos. Situando la rosa de los vientos en una posición adecuada y ajustando el valor del radio para que abarque sólo la abertura se consigue optimizar la distribución de fotones. Para ello, hacer lo siguiente. Situar en el exterior y mover la rosa de los vientos para situarla en una posición adecuada. Seleccionar el *mr sun*. Ir a la sección *mr Sun Photons*. Activar *Use Photon Target*. Con esto aparecerá un cilindro magenta en torno a la línea de conexión entre el sol y la rosa de los vientos (quizás no se vea inicialmente por ser el radio demasiado grande o demasiado pequeño). Modificar el radio para que se adapte aproximadamente a la abertura por la que se quiere que entren la mayoría de los fotones.

Como puede comprobarse si se utiliza este recurso, se reducirá el tiempo de cálculo pues sólo se enviarán fotones a través del cilindro virtual definido por este parámetro.

### Control de la fuga de color (*color bleeding*) con fotones

Durante el transporte de fotones se computa la pérdida de energía con la distancia, además de la absorbida o reflejada por cada superficie que encuentra en su camino. También se computan las transferencias de color

de una superficie a otra lo que afecta al color de las superficies, a lo que se denomina a veces “sangrado de color” (*color bleeding*). Este efecto puede resultar perceptualmente excesivo en algunos casos, debido a que nuestro sistema visual aprende a “descontar” determinados efectos de reflectancia, y hay recursos especiales que permiten corregirlo si se considera necesario.

La fuga de color se puede controlar por medio de un *shader* especial, el *mr Raytype Switcher* que se describe en el libro de Simulación de Materiales. Ver los ejemplos que se dan allí si interesase controlar este efecto.

### Fotones con V-Ray

Con *V-Ray*, al igual que con *mental ray* o cualquier otro programa de simulación que utilice fotones, el cálculo por Fotones es adecuado como segundo motor pues por sí mismo no puede llegar a resultados sin ruido en tiempos razonables. Tiene la ventaja, como ya hemos visto, de que es bastante rápido y puede rastrear toda la escena por lo que es una herramienta muy adecuada para escenas con rincones a los que no llega la luz directamente.

Y en general se utiliza así: como motor secundario, en combinación con *Irradiance*, de modo similar a *Final Gather* de *mental ray* que también conviene combinar con fotones si hay muchos rebotes.

En los primeros apartados que siguen lo utilizaré sólo como motor primario para que se entienda mejor el sentido de los parámetros principales. Y en el apartado final de esta sección veremos como se utiliza en combinación con *Irradiance* que, como decía, es la opción más corriente y más recomendable.

### Parámetros generales

Si, en la sección *Indirect illumination* (GI) de las *Options* de *V-Ray*, se escoge como *Primary Engine*, la opción *Photon Map*, los parámetros de la siguiente sección cambiarán para mostrarnos los parámetros generales de este método. Doy una descripción muy sinté-



ticas, con los valores predeterminados entre paréntesis. Y en la siguiente sección veremos con más detalle los principales.

El grupo *Basic Parameters* incluye los siguientes. *Max Density* (10) y *Max Photons* (30) son patrones principales que se detallan más adelante. *Max Bounces* (10) controla el número de rebotes de la luz y depende de la



Figura 6.26 Fotones con V-Ray. Valores predeterminados 10/30 (*max Density/max Photons*), 4 bounces. Tiempo 1X (de referencia para los siguientes)

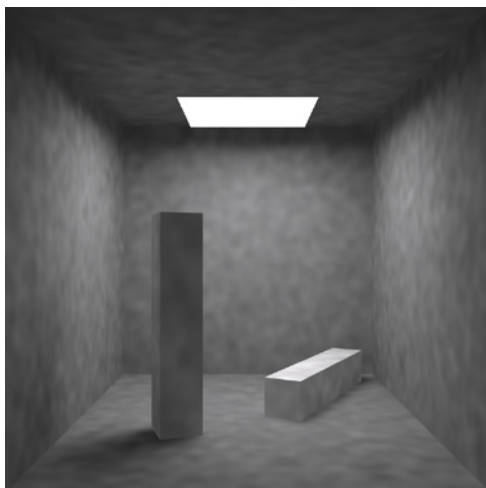


Figura 6.27 Fotones con V-Ray. Variación de *max Density*: 0.5 con *max Photons*: 30. Tiempo 1.3X

complejidad de la escena pues sólo se necesitará aumentarlo en escenas con muchos rincones. *Retrace bounces* (10) está relacionado con el parámetro anterior y con el siguiente y especifica cuantos rebotes se calcularán por métodos más eficaces (variantes de Montecarlo) en superficies críticas. Suele ser igual al número de *bounces* y no tiene efecto si el siguiente parámetro, *Retrace corners* (0.0), se deja en su valor predeterminado, 0.0 que es lo mejor que se puede hacer si no hay superficies críticas, como esquinas o superficies con cambios bruscos. En caso contrario, este valor (propio de V-Ray, no de unidades de escena) indica que se rehará el cálculo para este tipo de superficies (lo que mejorará el resultado a costa de mayor tiempo). La opción dada por la casilla *Override Search Distance* (desactivado) bloquea o desbloquea el uso de *Search dist* (1e+08), un parámetro que permite especificar manualmente el ámbito de búsqueda de los fotones y que está estrechamente relacionado con la densidad del mapa de fotones. Si se aplica, conviene que sea del orden de, como mínimo, el doble (tres o cinco veces más puede ser mejor) del valor de *Max Density*. Como la interpolación se efectúa a partir del cálculo previo, en el momento de hacer un *render*, no es necesario rehacer el cálculo si se cambia este valor. Valores bajos acelerarán el cálculo a costa de más ruido y viceversa. *Multiplier* (1.0) tiene el significado habitual pero en este caso afecta a la intensidad del mapa de fotones, si bien esto repercutirá en la iluminación global. *Convex hull area estimate* (desact) puede ayudar a suavizar el contraste excesivo en esquinas (a costa de mayor tiempo de cálculo. *Store direct light* (act) hace que se tenga en cuenta la contribución de las luces directas en el cálculo. Si no se activa, la contribución se deriva del cálculo con lo que el resultado puede ser más adecuado para luces difusas pero a costa de mayor tiempo.

El grupo *Mode* permite elegir entre *New map* (activado), para crear un nuevo mapa cada vez que se hace un *render* o *From file*, que utiliza un mapa previamente grabado. Si

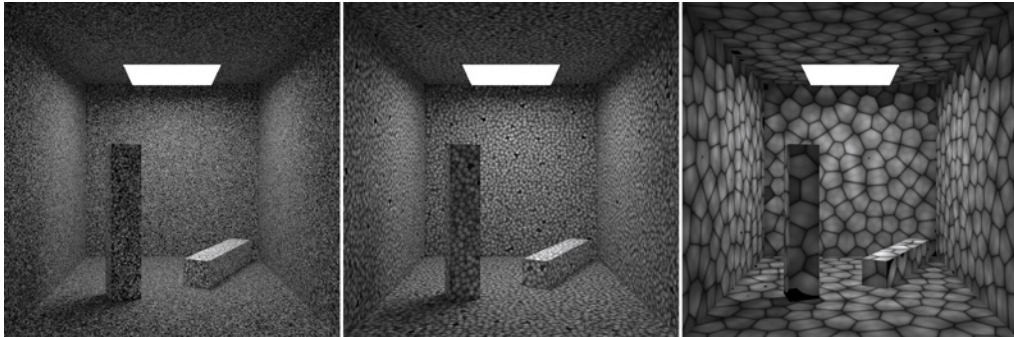


Figura 6.28 Fotonos con V-Ray. Diferencias, con max Photons 2, de variar la max Density. : a) 0.5, tiempo 7.9X, b) 2.0, tiempo 3.5X, c) 10.0, tiempo 1.4X



Figura 6.29 Fotonos con V-Ray. Diferencias, con max Density 8, de variar el valor de max Photons: a) 2, tiempo 1.6X, b) 4, tiempo 1.4X, c) 8, tiempo 0.9X, d) 16, tiempo 0.9X

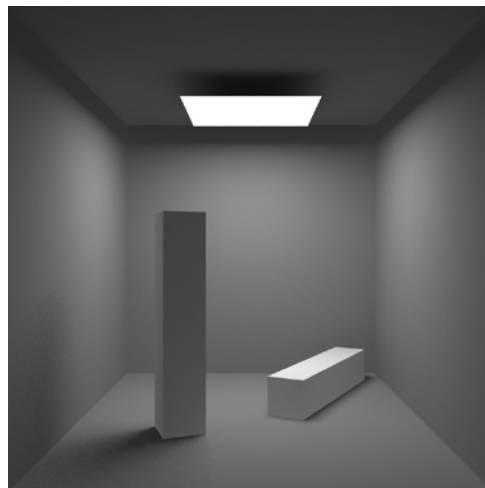


Figura 6.30 Combinación de irradiancia y fotonos con V-Ray. Valores predeterminados para irradiancia (Min/Max -3,-1; HSph 50, Samples 20) y fotonos (Bounces 10, Density 10, Max Photons 30). Los resultados son similares a los obtenidos con solo irradiancia pero algo más luminosos. El tiempo es ligeramente superior, del orden de 1.2X aunque tanto los resultados como el tiempo variarán mucho según los casos





los valores que se van a modificar no requieren rehacer el cálculo es conveniente grabar el resultado. Por ejemplo, supongamos que hemos obtenido un resultado más o menos satisfactorio con una densidad de 10.0 pero queremos ajustar el valor de *Search density* (que no requiere rehacer el cálculo). Presionar el botón *Save to file* después de haber obtenido un resultado satisfactorio y grabarlo con un nombre adecuado en la misma carpeta en que esté el modelo que estamos trabajando. Luego modificar el valor de *Search dist* y cambiar la opción de *Mode* a *From File*. Presionar el botón *Browse* y escoger el archivo recién grabado cuyo nombre aparecerá en el cuadro de texto junto a *From File*. Al hacer un *Render* se comprobará que se ejecuta con mucha mayor rapidez pues sólo hay que ajustar los valores de interpolación de salida.

El grupo *Post Render* con las opciones “Don’t delete” (act), “Autosave”, etc., tiene el mismo significado que en *Irradiance*. Otro tanto ocurre con el grupo *Current Map*.

### Análisis

Los **parámetros principales** que habrá que ajustar adecuadamente en la gran mayoría de los casos son *Max Density* y *Max Photons*.

a) **Max density** (0.0) controla el tamaño del mapa de fotones a través de la densidad del mapa. Antes de guardar un nuevo fotón en este mapa se comprueba si hay otros fotones dentro de la distancia especificada por este parámetro. Si los hay, no se guarda un nuevo fotón pero se ajusta la energía almacenada por los que ya existen. Y si no, se guarda un nuevo fotón en el mapa. Este modo de gestión del mapa permite hacer un uso más eficiente del tamaño del mapa pues no se requiere dar un número global, ya que el parámetro de densidad se encarga de adaptarlo a la escena. Si el valor de este parámetro es muy pequeño la densidad será mayor y habrá más fotones y la calidad será mayor pero el tiempo de cálculo también. Por encima de 0.0 el valor depende de las unidades y caracte-

rísticas de la escena. Un valor de 1.0, si las unidades son centímetros, significa que habrá una precisión de 1 cm con respecto al punto de intersección entre el rayo y el objeto (o 1 metro si las unidades son en metros). Una forma de estimar el valor inicial adecuado es medir dos objetos cercanos que deban recibir una iluminación semejante y hacer el valor de *Max density* igual a  $1/10$  de esta distancia. La modificación de este parámetro implica que hay que recalcular el mapa de fotones. Tener en cuenta que el valor predeterminado ha cambiado en las diferentes versiones de este programa por lo que la descripción anterior puede no coincidir con la que aparece en la versión que se esté utilizando.

b) **Max photons** (30) especifica el máximo número de fotones que se tendrán en cuenta para definir la iluminación en un punto. Es decir que si tenemos una densidad de 1.0 y una *Search distance* de 10, se tomarán todos los fotones que caigan dentro de una esfera de radio 10 (que, en principio, serían 10). Valores mayores suavizarán el resultado a costa de la precisión y viceversa. No es un parámetro demasiado importante a no ser que escojamos valores muy grandes de *Search distance* en comparación con *Max density* (para una densidad de 1 y una distancia de 50 si tendría sentido esta limitación). Pero si nos limitamos a una relación de  $1/5$  como máximo no se alcanzará el valor predeterminado.

Las figuras adjuntas ilustran el tipo de resultados que se obtienen variando estos parámetros principales.

### Combinación de irradiancia y fotones con V-Ray

Como ya he dicho más arriba, en el apartado sobre combinación de *Final Gather* de *mental ray* con fotones, el cálculo por fotones es muy rápido pero se necesita un gran número de fotones y es muy difícil, aún así, eliminar ruido, manchas e imperfecciones varias. Y por esto necesita ser combinado con algún tipo



de método de interpolación que elimine estos defectos.

En las últimas versiones de *mental ray* no se necesita aplicar esta combinación porque los métodos de FG ya llevan incorporados parámetros propios que computan los rebotes. Como ya he dicho, solamente en casos en que haya rincones a los que no llegue la luz o que requieran un gran número de rebotes merece la pena recurrir a esta combinación, que mejora el resultado y reduce los tiempos.

Sin embargo, en el caso de *V-Ray*, con sus sistema de motor primario y motor secundario, el método de irradiancia, por si solo, no computa adecuadamente los rebotes y requiere ser combinado con un motor secundario. Por esto, la configuración más corriente y que da buenos resultados en una gran mayoría de casos, es utilizar irradiancia como motor primario y fotones como motor secundario.

Si se hace así se obtendrán resultados relativamente rápidos, que correrán a cuenta de los fotones, sin necesidad de utilizar demasiados pues el ruido se absorberá con Irradiance. La última figura adjunta de este grupo muestra los resultados que se obtienen con esta combinación.

## Cáusticas

Cuando los rayos luminosos que se emiten desde un punto, atraviesan una lente o un medio no regular, forman un haz que no converge hacia un punto imagen sino que se dispersa en torno a una curva cuyas propiedades geométricas fueron estudiadas por primera vez en el siglo XVII (por Tschirnhausen, en 1682, pero también por Huygens y, posteriormente, por Lagrange, en el XVIII y Quetelet y Cayley en el XIX, entre otros).

La cáustica producida por un anillo, por ejemplo, es una evoluta de envolvente de círculo. Dada una curva  $C$  y un punto fijo  $P$ , se denomina catacáustica a la envolvente de los rayos que vienen de  $P$  y se reflejan a partir de la curva  $C$  y diacáustica a la envolvente de los rayos refractados. Los patrones de luz que se forman cuando una fuente de luz atraviesa

una botella o un vaso y cae sobre el mantel, o los que se forman en el fondo de una piscina o sobre su superficie, son ejemplos característicos de cáusticas.

Dicho de otro modo. Una cáustica es la envolvente de los rayos reflejados o refractados por una superficie curva o la proyección de esa envolvente de rayos sobre otra superficie. Esta envolvente es una curva o una superficie curva a la que cada rayo de luz es tangente, lo que define un contorno de luz concentrada. La figura adjunta muestra las cáusticas generadas por rayos paralelos que se reflejan sobre la cara interior de un círculo. Es una construcción geométrica que muestra claramente la envolvente y que es fácil de relacionar con las imágenes generadas por vasos o botellas o medios transparentes no planos.

Las cáusticas no se pueden procesar con los métodos corrientes que se basan, en general, en la *rendering equation* que se basa, a su vez, en el análisis de como se distribuye la luz directa en toda la escena a partir de su interacción con las diferentes superficies. El cálculo de iluminación global es afectado por las sombras, por zonas más oscuras que

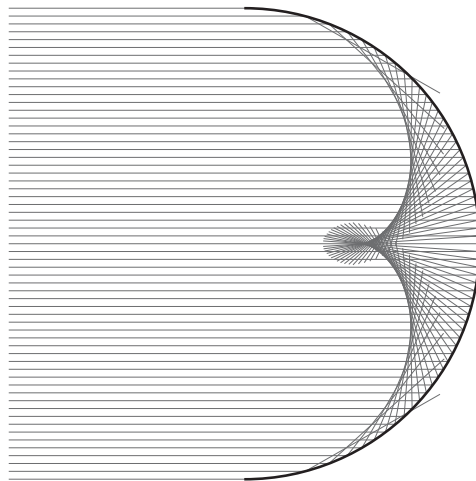


Figura 6.31 Cáusticas formadas por rayos paralelos reflejados por un círculo



otras. Pero lo característico de las cáusticas es que (en el caso de superficies transparentes, refractantes) añaden luz a las sombras a la vez que crean los dibujos característicos de la envolvente que he recordado más arriba.

Sin embargo sí es posible simularlas con métodos específicos, particularmente adecuados para generar estos efectos, como ocurre con los métodos basados en la emisión de fotones.

En arquitectura y diseño este tipo de efectos son poco frecuentes pero hay situaciones en las que pueden mejorar notablemente la calidad de la representación. Así ocurre con escenas en las que aparezcan piscinas, estanques, objetos metálicos o de cristal o, en general, objetos curvos muy reflectantes o transparentes y que sean claramente visibles.

## Caústicas con mental ray

### Descripción general

Para simular efectos cáusticos hay que dar una serie de pasos: designar los objetos que van a generar estos efectos, especificar que luces generarán efectos cáusticos y activar los parámetros correspondientes de iluminación avanzada. Concretamente, con *mental ray* habrá que hacer lo siguiente:

- Especificar los objetos que pueden **recibir** efectos cáusticos (por ejemplo una mesa que recibe los rayos que pasan a través de un vidrio). Para ello, seleccionar el objeto, entrar en el menú contextual (Botón Derecho del Ratón)/ *Object Properties* y en la sección *mental ray*, comprobar que está marcada la opción *Receive caustics* (que en principio lo estará pues no compromete a nada mientras no haya generadores de cáusticas).
- Especificar los objetos que pueden **generar** efectos cáusticos, por ejemplo, la esfera de las figuras adjuntas. Para ello, como en el caso anterior, seleccionar el objeto y, en *Object Properties/ mental ray*, marcar la

opción *Generate caustics* (que en principio estará desactivada).

- Especificar las luces que pueden generar efectos cáusticos y hacer otro tanto, seleccionándolas y, en sus propiedades, marcando la opción *Generate Caustics*.
- Comprobar que los materiales asignados a los objetos que van a generar objetos cáusticos tienen propiedades adecuadas de reflectancia o transparencia, tal como se indicará en los ejemplos que siguen.
- Entrar en *Render Setup* y, en la sección *Indirect Illumination/ Caustics and Global Illumination*, activar *Caustics* y especificar los diferentes parámetros tal como se describe en los siguientes apartados.
- Alternativamente, si se quiere controlar la generación de cáusticas localmente, editar la luz y, en la *mental ray Indirect Illumination*, activar la configuración manual, siguiendo también las indicaciones que se dan más adelante. Pero será preferible, en general, utilizar la configuración global.

Por último, tener en cuenta que no es fácil obtener resultados adecuados las primeras veces que se utilizan estos métodos por lo que conviene comenzar con modelos simplificados y avanzar poco a poco, desde configuraciones más simples a otras más exigentes.

### Parámetros generales

Al igual que con FG y fotones, la configuración del cálculo de cáusticas se ajusta desde *Render Setup/ Indirect Illumination/ Caustics and Global Illumination* (GI) que incluye 6 grupos que ya se han descrito en el apartado anterior, correspondiente a fotones. Los parámetros generales que se encuentran en esta sección son los siguientes. Para trabajar con cáusticas los principales son el primero (*Caustics*) y el anteúltimo (*Light Properties*).

El grupo *Caustics*, si se activa, muestra una serie de parámetros semejantes a los de los fotones. En primer lugar, un multiplicador y un modificador de color similares a los de fotones pero que, en el caso de las cáusticas,



tienen mayor importancia, sobre todo el *Multplier*, como veremos más abajo. Otro tanto cabe decir de *Maximum Num Photons per Sample*, *Maximum Sampling Radius* que también veremos con más detalle en los ejemplos que siguen. Si se deja activado *Opaque Shadows* las sombras resultan opacas y el cálculo es más rápido; si se desactivan, las sombras pueden ser parcialmente transparentes, en el caso de que interese que sea así, aunque el cálculo será más lento. Los *Filtros* son similares a los generales pero tienen mayor importancia en el caso de cáusticas. De las tres opciones, *Box*, *Cone* y *Gauss* son mejores las dos últimas que deberían ser las seleccionadas en la fase final. Es preferible utilizar *Cone* para aumentar la precisión de detalle y *Gauss* para suavizar el resultado. El tamaño del filtro afecta al área sobre la que actúa y es mejor dejarlo en su valor por defecto, 1.1.

El grupo *Global Illumination (GI)* se utiliza para fotones con iluminación global, como ya hemos visto, y no es necesario activarlo pues las cáusticas utilizan su propio sistema de fotones. El grupo *Volumes* se utiliza para cáusticas volumétricas que requieren que el material tenga asociado un *shader* especial tal como *Parti Volume* o, en general, un *shader* que procese lo que se denominan efectos de “medios participativos” (*participating media*) y que se utilizan para simular niebla, humo, polvo flotante en el aire, etc. Los parámetros correspondientes, número de muestras y radio, tienen el mismo significado que antes pero aplicados a las características de este tipo de *shaders*. Es un tema que se trata en el Libro de Simulación de Materiales. Los valores del grupo *Trace Depth* sirven para limitar el número de Reflexiones y Refracciones y también se utiliza principalmente con fotones. En general, para trabajar con cáusticas podemos olvidarnos de él. El último grupo, *Geometry Properties*, incluye una única opción, “All Objects Generate & Receive GI and Caustics” que nos ahorraría tener que activar esta opción para objetos particulares si realmente nos interesa que todos los objetos de la escena generen y reciban cáusticas lo que, en general, no será el caso.

El grupo *Light Properties* incluye dos parámetros y es igual que el de fotones, con la única diferencia de que el principal parámetro se denomina *Average Caustic Photons per Light*. Esto permite establecer un valor inicial para todas las luces de la escena. El parámetro *Decay* tiene el mismo sentido que con fotones, especifica el rango de atenuación y, en la mayoría de los casos tampoco será necesario cambiarlo. Este valor afecta al exponente con que se afecta a la distancia: un valor de 0.0 significa no atenuación, uno de 1.0 atenuación lineal y un valor de 2.0 atenuación cuadrática inversa que es el valor físicamente correcto.

Si editamos las luces que generan caústicas, en la sección *mental ray indirect illumination*, se puede elegir entre dos opciones:

a) Calcular automáticamente la energía y los fotones (*Automatically Calculate Energy and Photons*). Esta es la opción predeterminada que incluye tan sólo unos multiplicadores para tres parámetros (*Energy*, *Caustic Photons* y *GI Photons*) cuyo valor inicial es 1.0. Si se dejan estos valores como están, los cálculos dependen de la configuración global aunque pueden modificarse para la luz concreta que se esté utilizando (es decir que si se da al multiplicador *GI Photons*, por ejemplo, un valor de 2.0, se multiplicará por dos el número de fotones que envía esta luz);

b) Calcular de modo independiente los valores de esta luz (*Manual Settings*). Si se activa esta opción los valores de cálculo se establecerán de modo independiente para la luz seleccionada. En este caso los valores predeterminados para los tres parámetros anteriores son los siguientes: *Energy* 500.000, *Caustic Photons* 10.000, *GI Photons* 10.000. Y habrá un cuarto parámetro que es el valor de atenuación (*decay*) que inicialmente es 2.0 y que tiene el mismo sentido que con la configuración global. El parámetro *Energy* no afecta a la intensidad de la escena sino a la proporción de energía que cada fotón transmite en su recorrido.

Tener en cuenta que si se utilizan a la vez cáusticas y fotones (GI) una luz que nos in-



terese tan sólo para generar cáusticas, no para contribuir a la iluminación de la escena, debería editarse en modo Manual y establecer su valor de *GI photons* en 0.0. Otra posibilidad es seleccionar la luz y hacer que no contribuya a la iluminación difusa bajando su intensidad 0.0 de modo que solo arroje fotones caústicos.

### Análisis

Los **parámetros principales** son similares a los que hemos visto para fotones por lo que los resumo más brevemente. Pero los criterios de uso son diferentes. En el caso de los fotones utilizados como base de cálculo para la iluminación global nos interesa suavizar los resultados, por lo que es conveniente usar radios relativamente grandes y numerosos rayos que atenúen la transición. En el caso de cáusticas nos puede interesar, hasta cierto punto, lo contrario: generar patrones bien definidos que dibujen con más precisión la cáustica. Naturalmente esto se hace a costa de más tiempo de cálculo y más tiempo dedicado a buscar el valor adecuado que puede moverse en un rango bastante estrecho, por lo que hay que encontrar un equilibrio entre estas exigencias que, como siempre, dependerá de cada caso.

Para ganar tiempo merecerá la pena encontrar un valor adecuado cuanto antes para el número de fotones y luego guardar el mapa con la opción de *Reuse* (ver el apartado correspondiente más adelante) para ahorrarnos este paso.

a) **Número de fotones** (*Average GI Photons per Light*). Igual que con fotones en el caso general, sin cáusticas. Pero los valores adecuados serán inferiores pues no se aplican a la totalidad de la escena. En muchos casos bastará con 50.000 o 100.000. Comenzar por valores del orden de los 10.000.

b) **Radio de los fotones** (*Maximum Sampling Radius*). Igual que con fotones. Si se aumenta este valor disminuirá el ruido y las

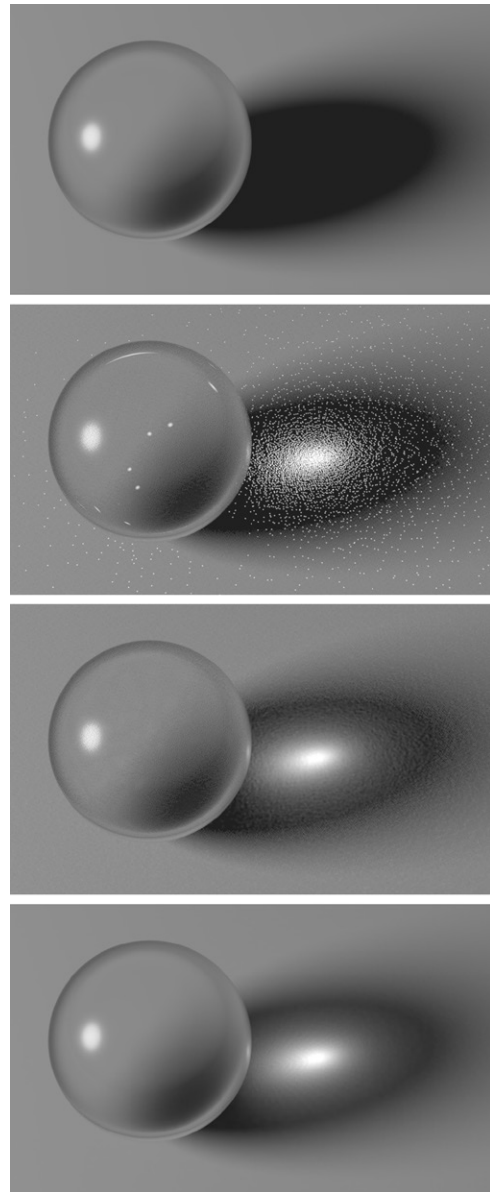


Figura 6.32 Cáusticas simples para una esfera sobre un plano. El multiplicador es 5.0 en todos los casos. Los valores corresponden a Radius/ Photons per sample/ Number of photons/ UV samples:

- a) Sin cáusticas, tiempo con uv 2x2, 1X,
- b) 0.1 / 1 / 10.000, uv 2x2, tiempo 1.2X,
- c) 0.25 / 100 / 100.000, uv 2x2, tiempo 1.3X,
- d) 0.5 / 100 / 500.000, uv 20x20, tiempo 9X



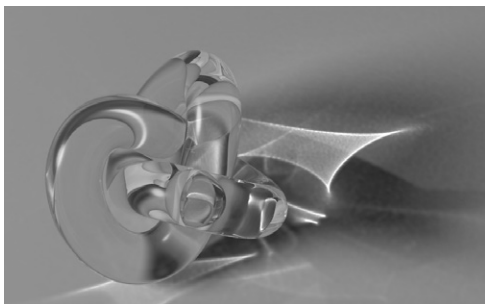


Figura 6.33 Cáusticas en un objeto complejo con la misma configuración que la figura anterior

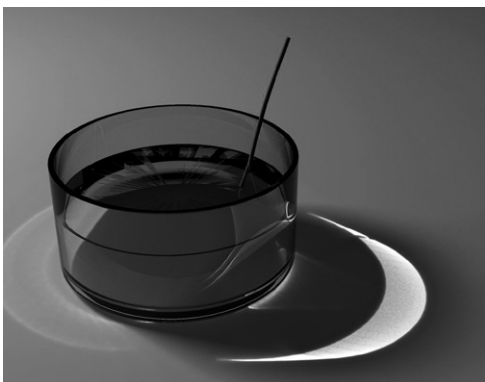


Figura 6.34 Cáusticas en un vaso de cristal con líquido de diferente IOR



Figura 6.35 Cáusticas reflejadas por el agua sobre un muro

imperfecciones pero también la precisión de las cáusticas. Si no se activa, el programa lo fija en un valor equivalente a 1/100 del área mayor de la escena. Dado que esta referencia es ambigua será preferible dar un valor concreto relacionado con el objeto que se va a iluminar, como en los ejemplos que siguen.

c) **Número de fotones por muestra** (*Maximum Photons per sample*). Igual que en el caso general: valores altos ocultan los defectos a costa de menor definición. Pero, de nuevo, en el caso de las cáusticas será preferible no utilizar valores demasiado altos para no perder precisión.

### Ejemplos

El primer ejemplo está referido a la figura adjunta que muestra una escena muy simple.

La escena consiste en una esfera de unos 10 cm de radio situada sobre un plano que se levanta con una suave curva por los extremos para envolver la escena de tal modo que se eliminen cambios bruscos que serían visibles en los reflejos.

La estructura del objeto afecta a la generación de cáusticas. En general, cuanto más rica y regular sea la malla mejores resultados obtendremos. Pero también aumenta el tiempo de cálculo. Por esto, siempre que sea posible, trabajar con un objeto paramétrico que permita modificar la densidad de la malla o bien con dos objetos de diferente resolución, utilizando el de menor resolución para hacer pruebas y substituyéndolo por el de mayor resolución al final.

El escenario (el plano curvado) tiene asignado un material mate, *Arch&Design*, de color gris hsv 0,0,0.65. Le he asignado algo de autoiluminación para que las sombras arrojadas no queden excesivamente oscuras. Para ello, en la sección *Self Illumination/ Glow* de este tipo de material, he activado la autoiluminación, asignándole una luminancia muy baja, de 5.0 cd/m<sup>2</sup>, pero suficiente para que las sombras no resulten negras.





En el caso del objeto que generará las cáusticas las posibilidades son variadas. Como ya he dicho más arriba, el tipo de material o combinación de materiales afectan de modo importante a las refracciones y los reflejos y, por consiguiente, a las cáusticas. Para comprobar los efectos básicos lo más sencillo es utilizar un material *Arch&Design* de *mental ray* y, en las plantillas predefinidas, escoger "Glass Physical" y dejar los valores predefinidos (*Reflectivity/Glossiness* 1.0/1.0, *Refraction* 1.0/1.0, *IOR* 1.5). Asegurarse de que en la sección *Advanced Rendering Options/Advanced Transparency Options* están marcadas las casillas "tratar object as...Solid" y "Refract light and generate caustic effect". También puede bajarse un poco el valor del *IOR (Index Of Refraction)* para que las cáusticas se concentren más.

Por lo que respecta a las luces, puede utilizarse todo tipo de luces pero resultará más sencilla la configuración si se utiliza una luz simple *mr area spot* como en este ejemplo en donde la única luz que hay en la escena es de este tipo. Está situada a unos dos metros del objeto y sus valores principales son: Intensidad (*Multiplier*) 0.8; *Hotspot/Falloff* 4.5/7.2, *Overshoot* activado (para que ilumine el escenario que rodea al objeto); *Area light parameters: type Disk, Radius* 50 cm, *UV Samples*: 20x20 (rebajar a 2x2 para hacer pruebas).

Una vez que se cuenta con un modelo de estas características, proceder como sigue.

- 1 Seleccionar el objeto que vaya a recibir cáusticas (el plano base) y hacer *BDR (Botton Derecho Ratón) /Object Properties*. En la sección *mental ray* comprobar que está marcada la opción *Receive Caustics*.
- 2 Seleccionar el objeto que vaya a generar cáusticas (la esfera) y hacer *BDR/Object Properties*. En la sección *mental ray* comprobar que está marcada la opción *Generate Caustics*.
- 3 Seleccionar la luz y hacer otro tanto: *BDR/Object Properties* y en la sección *mental ray* marcar la opción *Generate Caustics*.

- 4 Editar la luz y, para pruebas iniciales, en la sección *Area Light Parameters*, reducir el valor de *UV Samples* a 2x2. Luego, en la salida final, aumentar este valor a unos 20x20. Tener en cuenta que los tiempos se multiplicarán hasta cerca de diez veces más.
- 5 Entrar en *Render Setup/ Indirect Illumination*. Desactivar FG pues solo necesitamos los fotones de las cáusticas. Desactivar las cáusticas y hacer un *render* de prueba para comprobar que el resultado general es correcto. También convendrá en este estadio hacer una estimación de los valores adecuados de *UV Samples* para pruebas iniciales y finales. Así se obtendría un resultado como el de la figura adjunta (a).
- 6 Activar cáusticas. Cambiar la configuración como sigue para pruebas iniciales:  
Multiplier: 5.0  
Maximum Sampling Radius: 0.1  
Maximum Num Photons Sample: 1  
Average Caustic Photons per Light: 10.000
- 6 Analizar el resultado. Ir aumentando progresivamente los valores anteriores hasta que el resultado sea satisfactorio, como en las figuras adjuntas (b) y (c). En la fase final, editar la luz y aumentar el valor de *UV Samples*.
- 7 La configuración final de la figura (d) es la siguiente:  
Multiplier: 5.0  
Maximum Sampling Radius: 0.5  
Maximum Num Photons Sample: 100  
Average Caustic Photons per Light: 500.000

Al igual que hemos hecho con los fotones, también conviene comenzar con valores bajos que nos permitan visualizar los fotones para entender mejor como se distribuyen y como les afectan los parámetros. Si se empieza por fijar el radio en un valor muy pequeño, tal como 0.1, menos de un 10% del radio de la esfera, y el número de fotones por muestra en 1 y el número total de fotones en 10.000 o menos, veremos claramente como se distribuyen y podemos ir aumentando estos valores hasta encontrar una configuración razonable.



En algunas escenas puede merecer la pena aumentar bastante el valor del multiplicador para hacer más intensas las cáusticas. Algunos ejemplos de los propios desarrolladores de 3ds Max llegan a utilizar valores de hasta 100 para este multiplicador. Los valores de tamaño de radio y número de *samples* también pueden ser muy variables y dependen también de los casos y de las preferencias.

Como decía al comienzo, este apartado es sólo una introducción a los procedimientos básicos, desde el punto de vista de la configuración de la iluminación, pues el manejo adecuado de cáusticas requiere adentrarse en la simulación de materiales avanzados.

La siguiente figura adjunta se ha obtenido con la misma configuración anterior pero substituyendo la esfera por un objeto más complejo.

Como decía más arriba, los ejemplos más elaborados requieren utilizar materiales adecuados por lo que requieren adentrarse mínimamente en técnicas adecuadas de asignación de propiedades de materiales. Me remito por tanto, una vez más, al libro de Simulación de Materiales. Las otras dos figuras adjuntas muestran algunos ejemplos más complejos obtenidos con estas técnicas. Una de ellas representa un vaso con líquido que incluye dos objetos diferentes (el líquido y el cristal del vaso), con diferentes asignaciones de materiales que influyen en la configuración de las cáusticas. La otra representa una pequeña piscina en la que el agua también tiene asignado un material más complejo que se refleja sobre las paredes, formando dibujos caústicos.

### Problemas característicos

La simulación de efectos caústicos puede dar errores con facilidad. Aunque todas las técnicas de este libro se están introduciendo a través de ejemplos muy simples, para que nada distraiga de lo principal y se puedan reproducir con facilidad, esto es particularmente importante en el caso de cáusticas: convie-

ne comenzar por ejemplos sencillos antes de abordar casos más complejos.

En cualquier caso, es probable que aparezcan mensajes de error. Los más corrientes son:

- “*There are no caustic generators in the scene*”. Esto quiere decir que no hay luces que generen cáusticas. La solución es seleccionar la luz o luces y, en *Object Properties*, activar la opción “*Generate caustics*” que está desactivada por defecto,
- “*There are no caustic emitters in the scene*”. La solución es seleccionar el objeto u objetos que deben producir efectos caústicos y hacer otro tanto, ir a *Object Properties* y activar la opción de generar cáusticas,
- “*no photons stored after emitting 1000000 photons from light... [nombre luz]*”. Esto puede ser debido a varias causas pero lo más probable es que las propiedades del material sean inadecuadas: comprobar que sus niveles de reflectividad no son nulos pues esto impide que genere rayos derivados.

Si se cumplen todos estos requisitos el programa no emitirá mensajes de error pero es entonces cuando hay que comenzar a modificar los parámetros para que el resultado sea el deseado.

Si aparecen puntitos dispersos alrededor del objeto esto es debido a que los fotones son visibles. En general se eliminará si se aumenta el tamaño del radio o el número de fotones por muestra o ambas cosas.

Si las cáusticas son poco visibles las soluciones son las ya mencionadas: aumentar la calidad del procesamiento (más fotones) e intentar encontrar el punto de equilibrio para el radio y el número de muestras. También hay que probar a aumentar el valor del multiplicador que intensifica la luminosidad de la cáustica sin afectar a la iluminación de la escena.

Si las luces son demasiado suaves o demasiado intensas se puede ajustar su intensidad para cáusticas por medio del multipli-



cador, lo que facilitará volver a reajustarlas. En una escena con varias luces, seleccionar sólo las luces que pueden generar efectos cáusticos y en la sección *mental ray Indirect Illumination*, cambiar los valores de multiplicadores globales que afectan a la energía, los fotones cáusticos y los fotones de iluminación global. De este modo se pueden seguir utilizando los parámetros generales de *Render Setup*. Si por alguna razón se prefiere aplicar un cálculo específico a cada luz, desactivar la casilla (“automatically calculate energy and photons”) que liga estas propiedades a las globales con lo que queda disponible el grupo de parámetros de cálculo (*Manual Settings*) para la luz seleccionada.

## 6.6 Métodos de optimización

### Reutilización de mapas de irradiancia, final gather y fotones

Al utilizar irradiancia (con V-Ray) o *final gather* (con mental ray) se genera un mapa de irradiancia (*irradiance map*) en el que se almacenan los puntos muestreados y los valores correspondientes. Si no se hacen cambios que afecten a sus valores, este mapa se puede grabar y, al repetir el cálculo, indicar al sistema que utilice el mapa en lugar de repetir el proceso para generarlo. Esto puede ahorrar un tiempo considerable en los cálculos.

Otro tanto ocurre con los mapas de fotones que también se pueden reutilizar siempre que los cambios no afecten al contenido del mapa. Si se graba un mapa de fotones nos ahorramos la primera fase, la de creación de la estructura del mapa y de almacenamiento de los fotones. Durante la segunda fase (de *render*) todavía se puede ajustar el modo en que se interpolan los fotones, qué radio se aplicará y cuantos fotones intervendrán en este promedio.

Sin embargo es importante tener claro cuales son los cambios que afectan al mapa y cuales no. Por desgracia estos cambios son,

a menudo, frecuentes, por lo que este recurso es limitado.

Las limitaciones son similares para irradiancia y para fotones y se resumen a continuación. El mapa se debe reconstruir si:

- Se ha cambiado la geometría de la escena o las propiedades de los materiales.
- Se ha cambiado la posición de las luces. Como la luz directa y la indirecta se calculan de modo independiente, es posible cambiar parcialmente la configuración (no la posición) de la luz directa sin necesidad de cambiar el mapa de fotones que solo depende de la indirecta.
- Se ha cambiando la configuración de irradiancia o FG (densidad, número de rayos, valores de interpolación, rebotes, trace depth, etc.).
- Se ha cambiado la configuración del *Photon Map* (número de fotones, fotones por punto, trace depth, etc).
- Si se mueve la cámara esto afecta de diferentes modos a irradiancia, FG y Fotones. Como los mapas de irradiancia y FG se genera desde la cámara el mapa se modifica. Pero tanto en V-Ray como en mental ray no es necesario reconstruir todo el mapa pues se pueden añadir datos en modo incremental. En V-Ray, bastará con seleccionar la opción “Incremental add to current map” en el grupo *Mode* de *Irradiance*. Y en *mental ray*, la opción “Incremental Add FG Points to Map Files” en el grupo *Final Gather Map* de la sección *Reuse....* Por otro lado, como el mapa de fotones se genera desde la luz se puede cambiar la posición de la cámara sin necesidad de reconstruir el mapa de fotones.
- El cambio de resolución de salida no afecta al mapa, por lo que un procedimiento relativamente corriente es generar el mapa para resoluciones menores y luego usarlo para mayores. Pero dado que más resolución implica más muestras será más que conveniente generar el mapa a baja resolución con valores altos de configuración de FG y, en su caso, fotones.



Si estas condiciones se cumplen, el procedimiento de reutilización es sencillo y lo describo en primer lugar para **mental ray**. En la sección de *Global Illumination* del *Render Setup*, hay que abrir la última sección, *Reuse (FG and GI Disk Caching)*. Esta sección incluye tres grupos, *Mode*, *Final Gather* y *Caustic and Global Illumination Photon Map*. Los dos primeros se utilizan para FG y el tercero, que describiré más adelante, para fotones y cáusticas.

El grupo *Mode* incluye una lista colgante con tres alternativas: *Off* (predeterminada), *Single File Only* y *One file per Frame*. La primera desactiva la grabación, la segunda se utiliza con imágenes fijas y la tercera para animaciones (genera un archivo para cada marco). Utilizaremos por tanto la segunda. Al activarla aparece una única opción "Calculate FG/GI and Skip Final Rendering". Si se activa esta opción, al hacer un *render* se procesa solamente la primera fase y se graba el mapa, sin continuar el proceso hasta mostrar el resultado final. Pero podemos generar el mapa activando el botón correspondiente del siguiente grupo.

El grupo *Final Gather Map* incluye otra lista con tres alternativas: *Off* (predeterminada), *Incrementally add FG Points to Map Files* y *Read FG Points Only from Existing Map Files*. Utilizaremos la segunda para grabar el mapa y la tercera para leerlo. Al seleccionar esta segunda opción nos encontraremos con el nombre y la ubicación del fichero, si existe, y un botón para borrarlo. Presionar el botón (...) a la izquierda para cambiar el nombre y la ubicación si fuera necesario. Debajo aparece un botón para generar el mapa FG (*Generate FG Map Now*). Al presionar este botón se abre la ventana de *Render* y se lleva a cabo el primer pase que se graba en el archivo correspondiente. Cuando el mapa se ha grabado cambiar a la tercera alternativa, leer el mapa (*Read FG Points...*).

El grupo *Caustics and Global Illumination Photon Map* tiene la misma estructura que el anterior: tres alternativas para desactivar, leer o escribir, un nombre y una ubicación para el archivo en que se graba el mapa y un botón para generarlo.

El procedimiento característico, concretando todo lo anterior, es el siguiente.

1. Seleccionar, en el grupo *Mode*, la opción *Single File Only*.
2. Seleccionar, en el grupo *FG Map*, la opción *Incrementally add FG Points to Map Files*.
3. Comprobar que el nombre y ubicación del fichero en que se grabará el mapa es adecuado (mientras no se modifique el nombre y la ubicación del archivo entre este paso y el final, en que se lee el archivo, da lo mismo aceptar el nombre y ubicación dados por el programa).
4. Presionar el botón *Generate FG Map File Now* (alternativamente se puede marcar la opción, más arriba, "Calculate FG/GI and skip Final Rendering" para que al hacer un *render* se grabe sólo el mapa). Con esto se ejecutará la primera fase.
5. Modificar lo que sea, corrientemente, la calidad de salida del render.
6. Seleccionar, en el grupo *Final Gather Map*, la opción *Read FG Points Only from Existing Files*.
7. *Render*.

El volumen y el tiempo de cálculo son dos cosas a tener en cuenta para calibrar adecuadamente las configuraciones. El número de rebotes (*bounces*) afecta ligeramente al tiempo de cálculo y al volumen. El nivel de FG lo afecta notablemente. Y la resolución aún más. El siguiente cuadro resume algunos de estos valores para una escena corriente. El tamaño de resolución indica la anchura y la altura, iguales en estas pruebas. La letra "b" el número de *bounces*. El valor de referencia del tiempo (1X) es el de la configuración más baja.

FG	Tamaño archivo	Tiempo
Low, 512, 0	2.07 mb	1X
Medium, 512, 0 b	3.98 mb	2.75X
Medium, 512, 2 b	3.99 mb	6.25X
Medium, 512, 4 b	4.01 mb	8.00X
Medium 1024, 2 b	9.45 mb	15.00X
High, 512, 2b	17.48 mb	42.80X
High 1024, 2 b	17.55 mb	54.00X

Una vez que se cuenta con un archivo de tipo FGM se puede hacer un *render* sin pasar por esta fase, lo que acelera considerablemente el cálculo total. El tiempo de *render* es muy inferior al de generación de un mapa FG. La tabla adjunta muestra diferentes tiempos para diferentes valores de configuración de un mismo archivo, por referencia al tiempo de *render* para la misma configuración de referencia dada en la tabla anterior.

FG	Calidad	Tiempo
Low, 512, 0 b	1/4, 4	1X
Medium, 1024, 2 b	1/16, 1/4	1.4X
Medium, 1024, 2 b	1/4, 4	2.5X
Medium, 1024, 2 b	1, 16	6.8X

§ § §

El procedimiento, en **V-Ray**, es similar. En este caso, nos encontramos con lo siguiente.

Para grabar el mapa de irradiancia ir a *Options/ Irradiance Map*, en donde hay tres grupos que habrá que ajustar. En el grupo *Mode*, seleccionar la opción *Incremental add to current map*. Con esta opción si, por ejemplo, se cambia la posición de la cámara, los nuevos puntos que sean visibles a partir del cambio de posición se añaden al mapa previo en lugar de generarlo de nuevo. En el grupo *Post Render*, junto a *Autosave File*, seleccionar una ubicación y un nombre para el archivo (la ubicación predeterminada es la misma que la del archivo en que se está trabajando y el nombre predeterminado es "irr.vrmap"). Dejar marcadas las opciones "Don't delete" y "Autosave". En el grupo *Current Map* hay dos botones que también importan: *Save*, que graba el mapa actual, que se mantiene en memoria si está marcada la opción "Don't delete" y *Reset* que borra el mapa de la memoria.

Para grabar el mapa de fotones el procedimiento es prácticamente idéntico. Ir a *Options/ Irradiance Map*, en donde hay tres grupos semejantes. En el grupo *Mode* solo hay, en este caso, dos opciones: "New map" o "From file". Usar la primera opción para

crear un nuevo mapa y la segunda para leerlo cuando ya se haya creado. A la derecha, junto a *File*, hay un botón (...) para dar un nombre y una ubicación al archivo que se grabará con la extensión .vrmap) si no se quiere utilizar la opción "Autosave". En el grupo *Post Render*, junto a *Autosave File*, seleccionar una ubicación y un nombre para el archivo (la ubicación predeterminada es la misma que la del archivo en que se está trabajando y el nombre predeterminado es "irr.vrmap"). Dejar marcada la opción "Don't delete" y, si se prefiere, la opción "Autosave" que en este caso está desactiva por defecto. En el grupo *Current Map* hay dos botones iguales a los de irradiancia: *Save*, para grabar el mapa que se mantiene en memoria si está marcada la opción "Don't delete" y *Reset* que borra el mapa de la memoria.

## Uso de *importons* e *irradiance particles*

Tanto *importons* como *irradiance particles* no son métodos independientes sino que se incorporan a otros métodos para acelerar el cálculo. Los principios básicos ya se han descrito en el capítulo 3. Aquí doy un resumen mínimo e indicaciones concretas, referidas a las últimas versiones de 3ds Max y mental ray.

Los *importons* son similares a los fotones y, como estos recursos, son particularmente apropiados para rebotar en todos los rincones de la escena recopilando información lumínica. Sin embargo, a diferencia de los fotones no distribuyen energía sino que almacenan información cromática que indica el factor con que la iluminación de una determinada zona contribuirá a la imagen final. Y de ahí se puede derivar información sobre cómo deben distribuirse adecuadamente los recursos de cálculo. Otra diferencia importante es que, a diferencia de los fotones, que se emiten desde la luz, los *importons* se envían desde la cámara y rebotan en las superficies hasta alcanzar la luz. Con todo, con *mental ray* se utilizan los mismos tipos de *shaders* que los fotones por razones técnicas de implemen-





tación. Se envían antes de los fotones pues puede decirse que guían su distribución. El resultado final es que el mapa de fotones es mucho más pequeño que el que sería sin la ayuda de los *importons*.

Las *irradiance particles* se basan en *importons* y recogen la información captada por estos antes de que se inicie la fase de cálculo de iluminación propiamente dicha. Almacenan la cantidad de iluminación directa que llega a una determinada posición, de lo que deriva la denominación “irradiance”. También pueden almacenar, si se activa la opción correspondiente, la iluminación indirecta. La diferencia fundamental con el uso aislado de *importons* es que, al utilizar *irradiance particles* se desactiva el cálculo por fotones y el cálculo de iluminación avanzada se lleva a cabo por las *irradiance particles*. Otro aspecto importante es que se pueden utilizar para cálculos de tipo IBL, como los que veremos en la siguiente sección.

Aunque *mental ray* incluye soporte interno para utilizar *importons*, *irradiance particles* e *IBL*, estos recursos están ocultos (al menos desde la versión 2010 hasta la versión 2013) pues no han pasado suficientes pruebas de control. De ahí que, para utilizarlos, hay que buscar los archivos correspondientes, eliminar el símbolo de comentario que antecede a algunas líneas de programa y luego confiar en que funcionen adecuadamente. Pero la gran mayoría de usuarios preferirá seguramente utilizar plugins o scripts que facilitan esta tarea.

Para utilizar *importons* e *irradiance particles* con las versiones 2011 y 2012 había que instalar un plugin denominado *ctrlghost*. Este plugin se podía conseguir en [www.mentalray4design.com](http://www.mentalray4design.com) o haciendo una búsqueda por Internet con las palabras clave “importons” y “ctrlghost”. Los archivos incluidos en este plugin explican como utilizarlo por lo que me remito a ellos. En pocas palabras, nos encontraremos con que, en el panel *Create/Geometry/mental ray* habrá un nuevo botón, *mr Shader* que permitirá crear un *shader* auxiliar sobre la escena. Da igual la posición o el ta-

maño, solo importa que sea visible cuando se necesite para poder seleccionarlo y editarlo. Y al editarlo podremos acceder al *Material/Map Browser* con la sección de mapas en primer lugar y, al desplegar esta sección, se mostrará como mapa disponible para este *shader* “Ctrl.Ghost.Settings(ctrl.studio)”. Si se arrastra al Editor de Materiales se abrirá mostrando 7 secciones: *Final Gather*, *Importons*, *Irradiance Particles*, *Image Based Lighting*, *Progressive Rendering*, *Ambient Occlusion Cache* y *Others*.

Sin embargo también existe la posibilidad de acceder a funciones de *mental ray* via *scripts*. Y hay *scripts* recientes que creo que son preferibles al plugin citado pues son más sencillos de utilizar y proporcionan recursos ampliados. Aunque cualquier recomendación sobre estos temas corre el riesgo de quedar obsoleta, incluyo aquí algunas breves indicaciones para quien le interese.

Si se accede a la página <http://www.youcandoitvfx.com>, nos encontraremos (2014) con varios scripts desarrollados por Artur Leao para *mental ray* 3.10. Uno de ellos, *mr manager*, consiste en un *script* (de tipo macro, extensión .mzp) que incorpora recursos para utilizar conjuntamente *importons*, *irradiance particles*, *IBL* y, por añadidura *Unified Sampling*, la nueva técnica que ya he explicado en el apartado del mismo nombre del capítulo 4. La misma página da indicaciones sobre como utilizarlo por lo que tampoco me voy a detener en la descripción. Entre otras cosas porque estamos hablando de técnicas muy concretas de implementación de recursos generales y esto, como decía, puede cambiar con rapidez y, para cuando esto se publique, pueden haber aparecido otros recursos similares o el propio programa puede haberlos incorporado.

Para utilizar este *script* todo lo que hay que hacer es arrastrarlo sobre la ventana del programa y seguir las indicaciones de utilización. Nos encontraremos con recursos que substituyen con ventaja a alguna de las técnicas citadas, principalmente por lo que respecta a la rapidez.





## 6.7 IBL (*Image Based Lighting*)

### Procedimientos comunes a las diferentes aplicaciones

En el capítulo 3 hemos visto los principios teóricos generales en que se basa IBL. Y de esa descripción se desprende que, para utilizar este sistema de cálculo, necesitamos dos cosas: un mapa HDR adecuado y un programa que pueda procesar este mapa para obtener una distribución de iluminación que se aplique al modelo.

Lo primero, la obtención de un mapa HDR por diferentes métodos se ha descrito con detalle en el capítulo 4.

Lo segundo se desarrollará en esta sección. Como en casos anteriores, me concentraré en un programa principal, *mental ray* pero, al final, añadiré algunos comentarios sobre otros programas bastante utilizados en arquitectura y diseño. Hay que comenzar por recordar que la situación está evolucionando con rapidez y que cada vez más son los programas que cuentan con este tipo de recursos que son relativamente fáciles de implementar.

En los primeros apartados de esta sección me referiré a algunos aspectos técnicos que son relativamente independientes del tipo de aplicación.

#### Tipos de aplicaciones y tipos de mapas HDR. Conversión entre formatos

Desde el punto de vista de la proyección aplicada ya hemos visto que hay tres tipos principales de mapas que se utilizan como mapas de entorno: a) Equirrectangular (o Latitud/Longitud o Proyección Esférica), b) Cúbico (desplegado de diferentes modos, en cruz horizontal o en cruz vertical), c) Angular o Bola Especular (*Mirrored Ball* o *Light Probe*).

Y desde el punto de vista del formato interno también hemos visto que hay diferentes estructuras para lo que estamos denominando, genéricamente, “hdr”: *Radiance RGBE* (extensión .hdr), *Open EXR* (extensión .exr) de 16 y 32 bits por canal, *LogLuv* de 24 y 32

bits, *TIFF* de 32, 48 y 96 bits... Para no complicar innecesariamente la exposición, parto de la base de que solo utilizaremos los dos tipos más corrientes: los de extensión HDR y los de extensión EXR.

Para complicar un poquito más las cosas resulta que hay aplicaciones que solo aceptan formatos con la extensión HDR y otras que solo aceptan la extensión EXR de 16 bits y otras la extensión EXR de 32 bits.

Y esto puede requerir una conversión de formato. Como también hemos visto, una de las vías más recomendables es utilizar el programa *Picturenaut* que, entre otras cosas, permite abrir archivos en cualquiera de estos formatos y guardarlos en otro, directa o indirectamente pues para guardar un archivo EXR en formato EXR *half*, que no figura entre las opciones de guardado, hay que escoger como formato de salida TIFF. Así se abre un cuadro de diálogo, “Tiff save options” con tres opciones de profundidad de bits: 32 bits IEEE, 24 bits IEEE y *half* (OpenEXR) o EXR16.

Puede merecer la pena trabajar con EXR16 porque pesa menos, los resultados son similares y, si queremos editarlo en Photoshop, nos encontraremos con más herramientas disponibles.

Por último, también nos podemos encontrar, según las diferentes aplicaciones, con que al cargar un archivo en formato HDR o EXR hay que configurarlo adecuadamente. Más adelante veremos como se hace esto desde 3ds Max.

#### Combinación de mapas HDR y LDR para procesar fondos, entornos y reflejos

El uso de mapas de entorno plantea un problema general que se suma al problema particular ocasionado por el uso de mapas HDR como fuente de iluminación.

El problema general es que, si el mapa va a ser visible como fondo, la resolución tendrá que ser muy alta. Supongamos que tenemos una vista desde una cámara que abarque 45° en horizontal que es un campo de visión relativamente corriente. Esto quiere decir que,



para que la parte visible del mapa tenga una resolución adecuada, del mismo orden, el mapa debería tener una resolución 8 veces mayor. Si la resolución de salida fuese 1024, una resolución más bien baja que habría que duplicar o triplicar en muchos casos, el mapa debería tener una resolución de 8192 píxeles. Y si la resolución de salida es doble o el triple, habrá que multiplicar esta resolución del mapa global por 2 o por 3.

El problema particular es que, si se utiliza un mapa hdr como fuente de iluminación y, por añadidura, aparecen reflejos nítidos, las pequeñas imperfecciones del mapa se traducen en irregularidades que pueden resultar visibles. Por esta razón, como se ha comprobado con múltiples ensayos, suele ser preferible utilizar un mapa al que se ha aplicado un filtro de desenfoque para reducir imperfecciones. Pero si el mapa va a estar desenfocado no nos interesa que sea visible y tampoco es necesario que tenga demasiada resolución.

La solución inicial para sería, por tanto, utilizar dos mapas: un mapa HDR con poca resolución y desenfocado, aplicado por una proyección esférica, y que actúe como fuente de iluminación pero que no sea visible. Y otro mapa LDR, con mayor resolución, aplicado por una proyección plana o por una proyección esférica o cilíndrica con repetición sobre la zona visible, como fondo.

Pero si hay reflejos visibles nos encontraremos con que estos reflejan el mapa de fondo. Y si el mapa de fondo está preparado para una vista determinada no quedará bien en el reflejo. En estos casos, puede ser necesario utilizar un tercer mapa, si necesitamos reflejos precisos sobre algún objeto visible en primer plano y, por añadidura, utilizar *shaders* especiales que permitan indicar al objeto que no compute para el reflejo el mapa de entorno principal sino otro mapa de entorno secundario más adecuado a lo que nos interesa. Entre otras cosas porque la vista reflejada no tiene porque coincidir con la vista del fondo.

Todo esto requiere por consiguiente el uso de *shaders* especiales que permitan aplicar estos mapas a través de diferentes canales.

Como, entre otras cosas, esto depende de las diferentes aplicaciones y de los diferentes casos y, también, para que el proceso sea más fácil de asimilar, en los primeros cuatro ejemplos que desarrollaré más adelante me limitaré a un solo mapa. Y en el quinto y último ejemplo utilizaré tres mapas, uno para el fondo, otro para la iluminación dada por el entorno y otro para los reflejos.

Aunque esta última es la mejor opción, la descripción de las otras permitirá, no solo asimilar mejor el proceso sino adaptarlo a casos más sencillos.

### Entornos virtuales o integrados como objetos hemisféricos

En la mayoría de los casos los mapas de iluminación se aplican directamente a canales especiales que están disponibles en todas las aplicaciones, como mapas de entorno (*environment maps*). Pero hay algunos casos, con determinadas técnicas, en las que puede ser preferible utilizar un objeto geométrico concreto, un hemisferio o *sky dome* que envuelva la escena. La principal ventaja de este método es que el objeto sobre el que se proyecta el fondo está integrado en la escena y esto evita tener que hacer algunos ajustes sofisticados que son necesarios en los otros casos. Evidentemente, es más sencillo utilizar directamente mapas de entorno por lo que este recurso solo se utiliza en el caso de programas que no cuenten con los recursos que se describen más adelante o estos recursos no funcionan como uno quisiera.

Para ilustrar este caso, el primero de los ejemplos que siguen estará desarrollado con un objeto hemisférico integrado en la escena. En los siguientes utilizaré exclusivamente entornos virtuales, no integrados en objetos de la escena.

En cualquier caso, el mapa debe ser de tipo equirectangular, es decir, de proporción 2:1 y con los laterales ajustados de tal forma que coincidan al enrollarse. Esto debe ocurrir naturalmente si el mapa está bien generado pero no estará de más comprobarlo editan-



do el mapa en Photoshop y utilizando el Filtro *Displacement* (en menú Filtros/Otros) para desplazar la imagen a la mitad de la anchura y comprobarlo.

La escena debe estar modelada de tal forma que, en cualquier dirección, haya objetos que oculten el horizonte. En caso contrario veremos una línea recta artificial en el fondo que echará a perder la simulación. Hay otros recursos para evitar esto pero es preferible partir de la base de que la escena está auto-contenida pues los puntos de encuentro con el objeto contenedor siempre son problemáticos.

### Problemas característicos

Hay toda una serie de problemas que aparecen más o menos corrientemente cuando se ilumina un escenario virtual con mapas HDR. Aunque ya se indican en los casos que siguen, dado que es un tema importante y recurrente, resumo los principales en este apartado.

a) Resultados demasiado oscuros o demasiado claros. Esto puede tener diversas causas y lo primero que hay que comprobar es si el Control de Exposición es el adecuado. Un modo rápido de comprobarlo es desactivar el control de exposición. Si el mapa de fondo HDR resulta nítidamente visible estará claro cuál es la causa. También puede ser debido a que el archivo HDR tiene un *gamma* de 1.0. Esto podría solucionarse, en principio, de dos maneras. La primera manera sería cambiando la *gamma* de todas las texturas a 1.0 para procesarlas del mismo modo que el HDR. Pero resulta muy pesado y supone ir a contrapelo del flujo habitual, pues todos los programas de simulación están adaptados para trabajar con monitores corrientes y por tanto con *gammas* no lineales. Sin embargo, conviene echar un vistazo a las entradas que aparecen en Google cuando se introducen las palabras clave “xxx linear workflow” en donde “xxx” puede ser “3ds max” o “mental ray” o “v-ray”, etc. La segunda manera, por tanto, es aceptar que, por ahora, es preferible trabajar

en un espacio visual no lineal. Esto supone aplicar una corrección *gamma* al propio archivo HDR. Esto altera la distribución de iluminaciones, con lo que estamos abandonando el mundo de lo físicamente correcto para meternos en el mundo de lo aparentemente correcto. Pero esto nos permitirá seguir trabajando con nuestras texturas habituales aunque sea a costa de ciertas irregularidades, que con un poco de suerte serán imperceptibles. Hay que advertir que una corrección de 2.2 no funcionará: habrá que tantear un valor “adecuado” que oscilará en torno al punto blanco de la imagen y será más bajo, seguramente en torno a 1.5. Y habrá que ajustar también la exposición y la saturación de la imagen. Es decir, habrá que llevar ajustes similares a los que hemos visto en *Tone Mapping* pero que en este caso afectarán a toda la escena pues afectarán también a su iluminación. En cualquier caso, los detalles dependen de la aplicación y volveré a ellos en los ejemplos que sigue. Esto debe considerarse como una introducción general al tema.

b) Ruido. En muchos casos los resultados pueden mostrar ruido de diversos tipos: defectos, manchas, discontinuidades, etc. Esto es generalmente debido a falta de resolución del mapa. Como la gran mayoría de sistemas de GI se basan en trazar rayos a partir de muestras aleatorias, si las muestras caen en zonas poco representativas o yuxtapuestas, aparecerán discontinuidades que se traducen en algunos de los defectos mencionados. Hay dos soluciones para esto: la primera es mala y la segunda mejor. La primera consiste en aumentar la fuerza bruta, es decir, enviar más rayos para que haya más muestras y más rayos secundarios que reduzcan estos errores. El problema es que, en muchos casos, nunca lo reducirán del todo y los tiempos de cálculo irán creciendo exponencialmente. La segunda solución, ya indicada más arriba, es reducir la resolución del mapa HDR y aplicarle un filtro de desenfoque. Esto elimina las zonas de altas frecuencias que son las culpables de los artefactos que cuesta eliminar. Pues las zonas muestreadas estarán mucho más cerca



y serán más uniformes. Y esto es preferible hacerlo antes de cargar el mapa pues de este modo ahorramos memoria. Así que la solución es editar el HDR, reducir su resolución y aplicarle, por ejemplo, un filtro de desenfoque Gaussiano. Pero dado que esto afectará al fondo y los reflejos esto requerirá utilizar diferentes versiones del mismo mapa. También analizaremos con más detalle este problema con ejemplos concretos más adelante.

c) Un último problema es el sol o, en general, la presencia de zonas muy brillantes en la escena. Para empezar, hay que intentar que el sol no esté incluido en el HDR pues la diferencia de intensidades es excesiva y el rango se distorsiona. Pero por otro lado es una luz fundamental para incorporar al modelo de iluminación. Ahora bien, dado que la mayoría de los programas generan la iluminación global a partir de muestras tomadas del mapa HDR, el resultado más probable es que haya algunas muestras que correspondan a valores muy altos junto a otras muestras que correspondan a valores bajos. El resultado es que se producirán saltos bruscos en la representación de la iluminación que se traducirán en puntos claros y oscuros. Las soluciones varían con los programas pero, en general, se reducirán a dos principales: utilizar filtros que absorban estos defectos o editar el mapa HDR para eliminar o mitigar las zonas excesivamente brillantes. Otra posibilidad importante en la práctica es utilizar diferentes mapas para el fondo, la iluminación y los reflejos, como veremos en el último ejemplo. O bien utilizar una aplicación denominada sIBL ("smart IBL") a la que también volveré más adelante y que facilita el uso combinado de estos tres tipos de mapas.

### IBL con mental ray

El procedimiento general de generar un panorama y aplicarlo como mapa de fondo ya se ha visto en capítulo 4. Ver la sección 4.2 *Uso de imágenes HDR en cálculos de iluminación*. Allí se ha descrito como utilizar imágenes panorámicas de fondo, con independencia de

que fueran LDR o HDR y sin ligarlo a un procedimiento de cálculo específico.

Dando por supuesto que ya contamos con un mapa de estas características y de que conocemos el procedimiento para integrarlo en la escena (que se volverá a repetir brevemente) aquí nos concentraremos en el tipo de configuración adecuada para que la escena quede iluminada tan sólo con los valores dados por un mapa HDR.

### Carga y configuración de un archivo HDR o EXR

En todos los ejemplos que siguen hay que comenzar por cargar un archivo HDR desde el Editor de Materiales. A diferencia de lo que ocurre con un mapa de bits corriente, en este caso hay que comprobar o configurar el archivo de entrada.

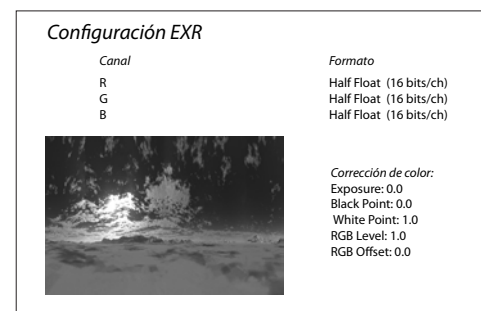
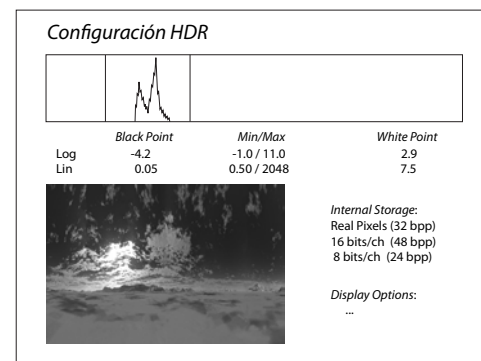


Figura 6.36 Configuración características de entrada de un archivo HDR y un archivo EXR



Estos archivos también se denominan, en la literatura técnica de los programas que los utilizan, *radiance files*, un término que se refiere a archivos con la extensión .hrd (o .pic) o .exr. Suele recomendarse utilizar preferentemente archivos .exr (formato *OpenEXR*) debido, entre otras cosas, a que incluyen soporte para canales alfa y son más fácilmente editables. También es importante que el motor de *render* esté configurado para manejar archivos con datos en coma flotante (32 bits por canal). En 3ds Max, por ejemplo, esto implica comprobar que el *Frame Buffer Type* sea de este tipo (que ya lo suele ser por defecto pero no está de más revisarlo desde *Render Setup/Renderer/Sampling Quality*).

Al seleccionar como Bitmap un archivo HDRi aparece el cuadro de diálogo habitual *Select Bitmap Image File*. Pero hay un botón, que corrientemente está inhabilitado y que ahora estará disponible: *Setup*. Al presionar este botón, se entra en un cuadro de configuración como el de la figura adjunta.

El cuadro muestra en la parte superior un histograma con la distribución de niveles de intensidad del archivo. Sobre el histograma habrá dos líneas verticales rojas que marcan la posición del punto negro, a la izquierda, y blanco, a la derecha. Debajo del histograma hay controles numéricos para ajustar el Punto negro (*Black Point*) y Punto blanco (*White Point*) con valores lineales y logarítmicos. Más abajo hay 3 opciones de almacenamiento: *Real Pixels* (32 bpp), *16 bits per channel, linear* (48 bpp) y *8 bits per channel, linear* (24 bpp). Ver la sección sobre formatos en el capítulo 2 si no se recuerda el sentido de estas opciones.

Puede dejarse la opción predeterminada *Real Pixels* (32 bpp) para trabajar con mental ray. Si, en este caso, se deja también activada la opción "Def Exp" (que sólo está disponible con *Real Pixels*) esto hace que la imagen se cargue tal cual es, sin aplicar ningún ajuste. Si se desactiva se pueden utilizar los parámetros del grupo *Exposure* para redistribuir los colores tal como se explica a continuación. Si se escogen las otras dos opciones, *16 bits per*

*channel, linear* (48 bpp) y *8 bits per channel, linear* (24 bpp), la opción "Def Exposure" queda anulada y lo habitual es utilizar los parámetros del grupo *Exposure*. Escoger la tercera opción (8 bits per channel) no tiene mucho sentido mientras que escoger la segunda es bastante corriente pues los resultados son suficientes y se sobrecarga menos el sistema.

Con *Exposure* disponible, es recomendable ajustar la posición del punto negro y punto blanco, siguiendo los resultados en el histograma y en la pequeña imagen incluida en el cuadro de diálogo, abajo a la izquierda. Esta pequeña imagen mostrará zonas magentas y cian (si no es así, cambiar los valores como se indica a continuación, para que aparezcan). Las zonas magenta corresponden a puntos cuyo valor de luminosidad es superior al máximo representable, al "White Point". Las zonas cian corresponden a puntos cuyo valor de luminosidad es inferior al mínimo representable, al "Black Point". Lo que más nos importará ajustar con precisión es la posición del punto blanco si bien la posición del punto negro afecta al rango global por lo que también conviene ajustarlo.

Para ajustar el punto negro, presionar las flechas del contador que hay junto a *Linear* (o *Log*). La línea roja se moverá hacia uno u otro lado según la flecha que estemos presionando. Si se desplaza hacia la derecha, hasta penetrar en el histograma, irán apareciendo marcas cian que irán aumentando a medida que penetremos más en la curva. Si ahora retrocedemos hacia la izquierda irán desapareciendo. Dejar la marca en el punto en que desaparecen: si se lleva más hacia la derecha estaremos recortando valores; si se lleva más a la derecha estaremos aplanando valores pues tendrán que repartirse en un área más extensa.

Para ajustar el punto blanco hacer lo mismo pero moviendo la marca que está a la derecha. Irán apareciendo marcas equivalentes de color magenta. Igual que en el caso anterior, dejar la marca en el punto en que desaparezcan no mucho más a un lado u otro.



Anotar el valor del desplazamiento lineal. Si, por ejemplo, la posición del punto negro está en 0.1... y la del blanco en 4.9... tendremos un valor de desplazamiento lineal aproximado de 5.0.

Al cargar la imagen, el Editor de Materiales mostrará, probablemente, una imagen demasiado oscura debido al desplazamiento de los valores. Con una imagen demasiado oscura difícilmente obtendremos resultados adecuados, sea cual sea su utilización pues no estaremos utilizando todo el rango dinámico de la imagen.

Si es así, una solución es editar el mapa y, en la sección *Output*, aumentar el *RGB level* en una cantidad similar a la dada por el desplazamiento lineal aunque esto puede depender de los casos y es posible que haya que jugar con valores de 5.0, 10.0 o más.

### § § §

Al cargar un mapa en formato EXR nos encontraremos inicialmente con un panel similar al anterior pero que incluye parámetros diferentes. El grupo *Gamma* incluye varias opciones, la que usará es "Use image's own gamma" (las otras dos opciones son "Use system default gamma" y "Override", que permite introducir un valor). Luego hay cuatro botones: a) *Devices* (opciones adicionales para leer el archivo con otros dispositivos), b) *Setup*, que es la opción principal que se comenta a continuación, c) *Info* (da información sobre la imagen), d) *View* (visualiza la imagen).

Al activar el botón *Setup* se abre otro panel con varias opciones que se resumen en la figura adjunta. Las principales son las que aparecen en el grupo *Color Correction* si se activa "Enable". En este caso se pueden modificar la Exposición (*Exposure*, 0.0), la posición de los valores extremos (*Black Point*, 0.0; *White Point*, 1.0) los niveles generales de valores (*RGB Level*, 1.0; *RGB Offset*, 0.0). En general no será necesario modificar estos valores, a diferencia de lo que ocurre con las imágenes en formato HDR, y se pueden dejar las opciones predeterminadas. Esto simplifi-

ca el trabajo y es una razón adicional para, en caso de duda, utilizar preferentemente el formato EXR.

### Alternativas de configuración para IBL

IBL se procesa, en principio, con *ray tracing*. Esto quiere decir que se adaptará bien a un sistema basado en *ray tracing* y que procese de modo natural la información dada por los objetos de la escena, sean mates o emisores. Y, por consiguiente, si se cuenta con un medio emisor, como puede ser un objeto que abarque toda la escena, no hay grandes dificultades para adaptar este tipo de organización al método general.

Sin embargo, los métodos de *irradiance* o *final gather* parten de un pase previo en que se computa la iluminación directa para, en un pase posterior, recoger la información derivada de este primer pase y redistribuirla adecuadamente. Esto quiere decir que si no hay algún sistema de iluminación previa o un objeto con propiedades de emisión especiales no podrá completar adecuadamente el proceso. Pero, por otro lado, ya hemos visto que el procesamiento de objetos iluminantes con FG es deficiente.

De ahí que, en principio, parece que contamos con tres posibilidades de aplicación de IBL: utilizar un sistema de irradiancia o FG apoyado en un objeto iluminante o en una luz celeste (*skylight*), utilizar un sistema de *ray tracing* / *path tracing* que procese directamente el mapa de entorno, utilizar métodos específicos de procesamiento de IBL proporcionados por la aplicación.

Es cierto que *Final Gather* tiene la capacidad de tomar muestras de valores de luminancia del entorno y aplicarlos a las superficies de la escena por lo que es un método que, en principio, se adaptaría bien a utilizar mapas de tipo HDR pues los rangos dinámicos ampliados pueden dar lugar a resultados más ricos en matices y en contraste. Sin embargo requiere usar luces complementarias pues no procesa bien los mapas como luces primarias. Por otro lado, el uso de luces adi-





cionales contribuirá a mejorar el resultado, como veremos en los ejemplos que se desarrollan más adelante. Convendrá alinear estas luces con las zonas más brillantes del mapa HDR utilizado para que la distribución general sea más coherente. Y, dado que interpola los resultados, habrá que tener especial cuidado en ajustar la configuración para prevenir defectos. Esto puede tener el inconveniente de que eliminaremos detalles. Si el mapa HDR incluye zonas muy brillantes (por ejemplo, el sol), la luminancia correspondiente a esos puntos será muy alta. Pero si la densidad de rayos enviados por FG es pequeña, tan solo unos pocos rayos alcanzarán la zona correspondiente del mapa. Al interpolar estos valores con otros cercanos pero muy bajos se producen saltos bruscos que se traducen en irregularidades de la imagen.

Una solución es reeditar el mapa HDR y rebajar la intensidad de esos puntos de valor muy alto. Esto puede hacerse desde fuera del programa, lo que permitirá hacer ajustes locales o desde dentro, utilizando un multiplicador que reduzca globalmente la intensidad. Este último recurso es más sencillo de aplicar pero puede ser insuficiente porque los altos contrastes se mantendrán.

Hay tres alternativas principales:

a) utilizar el filtro de *FG Noise filtering* (*speckle reduction*). Este filtro admite cinco valores 0 (*none*), *standard* (1), *high* (2), *very high* (3), *extremely high* (4). La ventaja de este método es que es más sencillo de aplicar. La desventaja, que oscurece toda la escena.

b) Editar el archivo HDR con un programa como *HDR Shop* para filtrar los valores altos e interpolarlos sobre una zona más amplia.

c) Editar el archivo HDR y superponer una zona gris oscura o negra sobre la zona de valores altos.

El uso de filtros, sin embargo, no sirve para reflexiones dispersas. Y para este problema hay que recurrir a los otros dos métodos. Pero hay otra alternativa, ya mencionada, que es utilizar diferentes mapas para la iluminación y para los reflejos.

Los tres primeros ejemplos que siguen se procesarán con estos dos métodos, que ya hemos visto: *final gather* (o *irradiance*) e *iray* (o bien, en otro programa, cualquier método alternativo de *ray tracing*/ *path tracing*).

El método que he denominado **IBL nativo** (de *mental ray*) en el cuarto ejemplo, se basa en recursos de *mental ray* (*IBL*, *Progressive Rendering*, *Unified Sampling*, *Irradiance Particles*, entre otros) que han permanecido ocultos pero accesibles desde la versión 3.8 de *mental ray* (disponible desde las versiones de 2011 para 3ds Max y Soft Image; en el año 2014 estaban disponibles la versión 3.11 de *mental ray* y en fase beta la 3.12). Esto quiere decir que se dejaban a disposición de los desarrolladores (que todo lo que tenían que hacer, en principio, era desocultarlos, eliminando el signo de comentario de las *libraries* adjuntas a los archivos de instalación) pero no a disposición de todos los usuarios pues no habían pasado por todos los protocolos de validación de Autodesk.

Aunque no hay demasiados detalles sobre la implementación parece que se basa en crear internamente un tipo de luz especial que procesa los valores correspondientes al mapa de entorno y los emite hacia la escena. Durante la fase de cómputo se premuestra el mapa de entorno para crear una estructura de datos que puede remuestrearse de modo adaptativo y cuyos valores se comunican a los *shaders* presentes en la escena como se haría en el caso de luces corrientes. IBL también puede combinarse con *progressive rendering* y *unified sampling*, dos nuevos recursos que se han descrito en el capítulo 4, en el apartado de *antialiasing*. También puede combinarse con *Ambient Occlusion* lo que mejora aún más los resultados.

En consecuencia han surgido en los últimos años varios plugins y scripts que permitirían utilizar estos recursos cargándolos manualmente en el programa. Algunos son muy interesantes pues utilizan IBL en combinación con *importons* e *irradiance particles*, además del nuevo sistema de *antialiasing Unified Sampling* que mejora y simplifica la elimina-



ción de defectos corrientes. Hay dos series de recursos adicionales que puede merecer la pena probar y que son muy similares. Uno, denominado *mr Optimizer*, se ha desarrollado por Thorsten Hartman y puede descargarse de <http://www.infinity-vision.de/page/shader-utilities>, aunque tiene el inconveniente que es de pago. Incluye los recursos de mental ray citados hasta la versión 3.10. Ver también [http://www.infinity-vision.de/page/ibl\\_mode](http://www.infinity-vision.de/page/ibl_mode).

El otro, *mr Manager 2013*, se ha desarrollado por Artur Leao y puede descargarse de <http://www.youcandoitvfx.com/tools/mr-options-manager/>. Es gratuito y su funcionamiento es muy simple, todo lo que hay que hacer es arrastrarlo sobre la aplicación 3ds Max y seguir las indicaciones de utilización.

Los parámetros incluidos tienen el siguiente significado: a) “Light” proporciona resultados precisos y converge hacia la solución correcta aunque es lento, sobre todo si se utilizan *shaders* con componentes semirreflectantes; b) “Automatic” es el método recomendado y proporciona resultados precisos a medio camino entre rapidez y calidad. Sus implementaciones internas puede cambiar con las nuevas versiones de *mental ray*. En las recientes utiliza métodos de *environment lighting cache*; c) “Approximate” da resultados más rápidos pero no exactos (*biased*). Está pensado para aplicaciones interactivas más que para imágenes estáticas de calidad. Implica el uso de *environmental lighting cache* y el modo de *environment lighting shadow* “solid”.

Si se cuenta con la versión 3ds Max 2014 puede utilizarse directamente IBL, que está disponible para la última versión de *mental ray*, la 3.11 (la 3.12 está a punto de salir en 2014). En la versión 2014 hay una nueva entrada en el *Render Setup* que incluye IBL. Ir a *Render Setup/ Global Illumination/ Skylights & Environment Lighting* (IBL). Esta sección incluye dos opciones “Skylight Illumination from Final Gather” y “Skylight Illumination from IBL”. Activar esta última. Configurar sus dos parámetros: *Shadow Quality* (rango 0.0 a 10.0, con 0.5 predeterminado) y *Shadow*

*Mode* (*Transparent*, más precisa y más lenta o *Opaque*, menos precisa y más rápida).

Tener en cuenta que el uso anterior implica dos cosas: a) que se ha creado previamente una *Skylight* a la escena y que, en el grupo “Sky color”, se ha seleccionado “Use Scene Environment” (aunque también es posible utilizar un mapa independiente y más de una luz con diferentes mapas), b) que se ha asignado a la escena como mapa de entorno un mapa adecuado, preferiblemente de tipo HDR o EXR.

Si se utilizan estos recursos todo lo que hay que hacer es añadir el mapa HDR como mapa de entorno y activar *render* (desactivar antes todas las luces de la escena. Si estuvieran activas las luces predeterminadas de 3ds Max crear una luz cualquiera y apagarla para anular las predeterminadas). No se requiere ni *FG*, ni *iray* ni *skylights*. Aparte de esta ventaja, los resultados son de mayor calidad que los obtenidos con FG en los ejemplos previos.

### § § §

Por otro lado, si utilizamos un mapa de entorno para que funcione, a la vez, como mapa de iluminación de entorno y como mapa de fondo, nos encontraremos con requisitos distintos. Para el mapa de iluminación necesitaremos un mapa equirectangular, de proporción 2:1, con suficiente resolución para abarcar adecuadamente toda la escena pero con poco detalle pues, para prevenir defectos debidos a insuficiente muestreo, es preferible

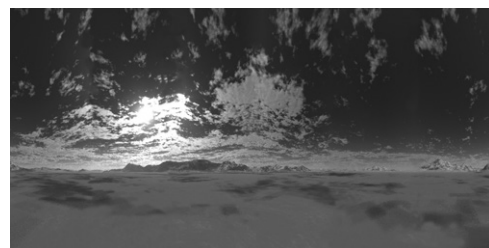


Figura 6.37 IBL. Mapa (en versión LDR) del archivo HDR generado en Terragen mediante los procedimientos descritos en el capítulo 4 y que se utilizará en los ejemplos que siguen

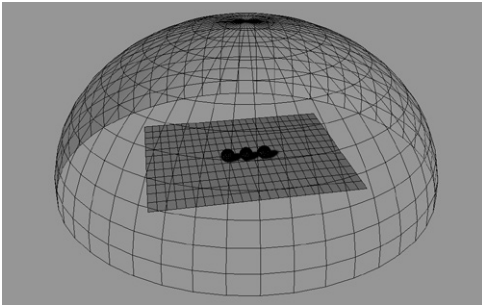


Figura 6.38 IBL. Ejemplo 1. Hemisferio envolvente. Vista general de la escena

que esté ligeramente desenfocado para evitar zonas de altas frecuencias que se traducirán en ruido por muestreo insuficiente. Para el mapa de fondo necesitaremos un mapa con suficiente detalle pero con una resolución que no tiene que ser mayor que el tamaño de salida si vamos a adaptarlos al fondo de pantalla. Y, por añadidura, si los objetos incluyen reflejos, necesitaremos mapas que tengan una resolución intermedia.

Resumiendo. Nos encontramos con los siguientes casos principales:

a) **Un mapa** aplicado a

i) un objeto hemisférico que cubre toda la escena. Esta es la opción más antigua pero que puede seguir siendo útil en determinados casos y puede ser la única alternativa para determinadas aplicaciones. Es una solución que funciona correctamente con irradiancia o FG, aunque los resultados tienen menos calidad que con FG combinado con una *skylight* y bastante menos calidad que con IBL nativo.

ii) una *sky light*. Es una solución que también funciona correctamente con irradiancia o FG aunque los resultados tienen menos calidad que con IBL nativo.

iii) el entorno (*environment map*) como único recurso. Esta es la solución más sencilla y esta es su principal ventaja. Sus dos principales inconvenientes son que es muy lenta y que es poco flexible pues prácticamente no admite ajustes.

b) **Dos mapas**: un mapa HDR aplicado como fuente de iluminación, como en los casos anteriores y otro mapa LDR aplicado de modo independiente como fondo.

c) **Tres mapas**: los dos primeros, como en el caso anterior, para la fuente de iluminación y para el fondo y el tercero para los reflejos en caso de que haya objetos reflectantes visibles desde la cámara.

### Escenario y mapas utilizados en los cuatro primeros ejemplos que siguen

Para facilitar la comparación, en todos los ejemplos que siguen utilizaré el mismo escenario, muy simple: un plano de unos 300x300 cm situado en el centro de la escena. Sobre este plano están situadas tres esferas de 10 cm de radio.

Las tres esferas y el plano tienen asignados un material de tipo mental ray *Arch & Design* de color gris claro. En la primera por la izquierda el material es puramente especular, en la segunda es semiespecular, con valores de reflectividad 0.65, 0.35 (reflectancia, glossiness), en la tercera es mate. El plano es casi mate, con valores de reflectividad 0.35, 0.15.

En todos los casos utilizaré también un mismo mapa de entorno en dos versiones, LDR y HDR. El mapa se muestra en la figura adjunta y es el que he generado en Terragen con los procedimientos descritos en el capítulo 4. Para utilizarlos en los ejemplos que siguen, hay que tener dos mapas similares preparados en una carpeta adjunta al archivo de la escena. Luego hacer lo siguiente: a) Entrar en el Editor de Materiales, b) Crear un mapa JPG. Para ello, seleccionar un visor, abrir el *material map browser* y escoger un mapa de tipo bitmap. Con esto se entrará en el navegador de Windows. Seleccionar el mapa de tipo LDR, c) En la sección *Coordinates*, seleccionar la opción *Environment* (en lugar de *Texture*) y como tipo de *mapping*, *Spherical Environment*. Cambiar el *tiling* a -1.0 (en lugar de 1.0) para que la orientación del mapa se invierta, d) Crear un mapa de tipo EXR repitiendo los mismos pasos que para el anterior.



Revisar la sección sobre “Carga y configuración de un archivo HDR o EXR para configurar adecuadamente este tipo de mapa.

Y en todas las escenas pueden entrar todos o parte de los tres recursos principales que ya hemos visto en el capítulo 5 ligados a los sistemas de luz diurna: el *mr Sun*, el *mr Sky* y el *mr Physical Map*. En principio el sol (*mr Sun*) está desactivado pero podría utilizarse, reduciendo su intensidad por medio del multiplicador, para complementar los efectos de sombra. El uso de *mr Sky* puede estar activado y ligarse a un mapa HDR a través

del tercero, el *mr Physical Map*, si se usa FG como sistema de cálculo. Los valores de estos tres *shaders* están relacionados, en principio, lo que asegura su consistencia.

En lo que sigue me referiré a estos mapas de entorno como “el mapa LDR” y “el mapa HDR”.

### Ejemplo 1. Un mapa. Con FG y un objeto envoltente hemisférico irradiante

Ya hemos visto que, en muchos casos, puede ser preferible utilizar un objeto geométrico

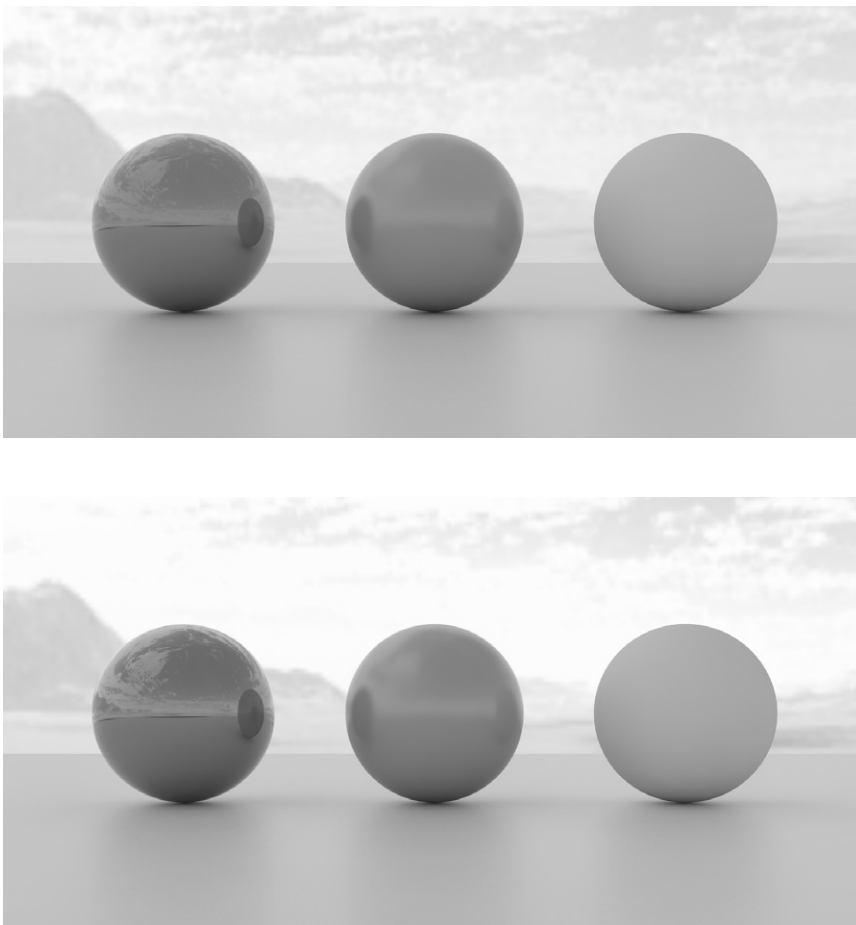


Figura 6.39 IBL. Ejemplo 1. Hemisferio envoltente: a) Resultado con un mapa LDR (jpg), b) Resultado con un mapa HDR (exr)

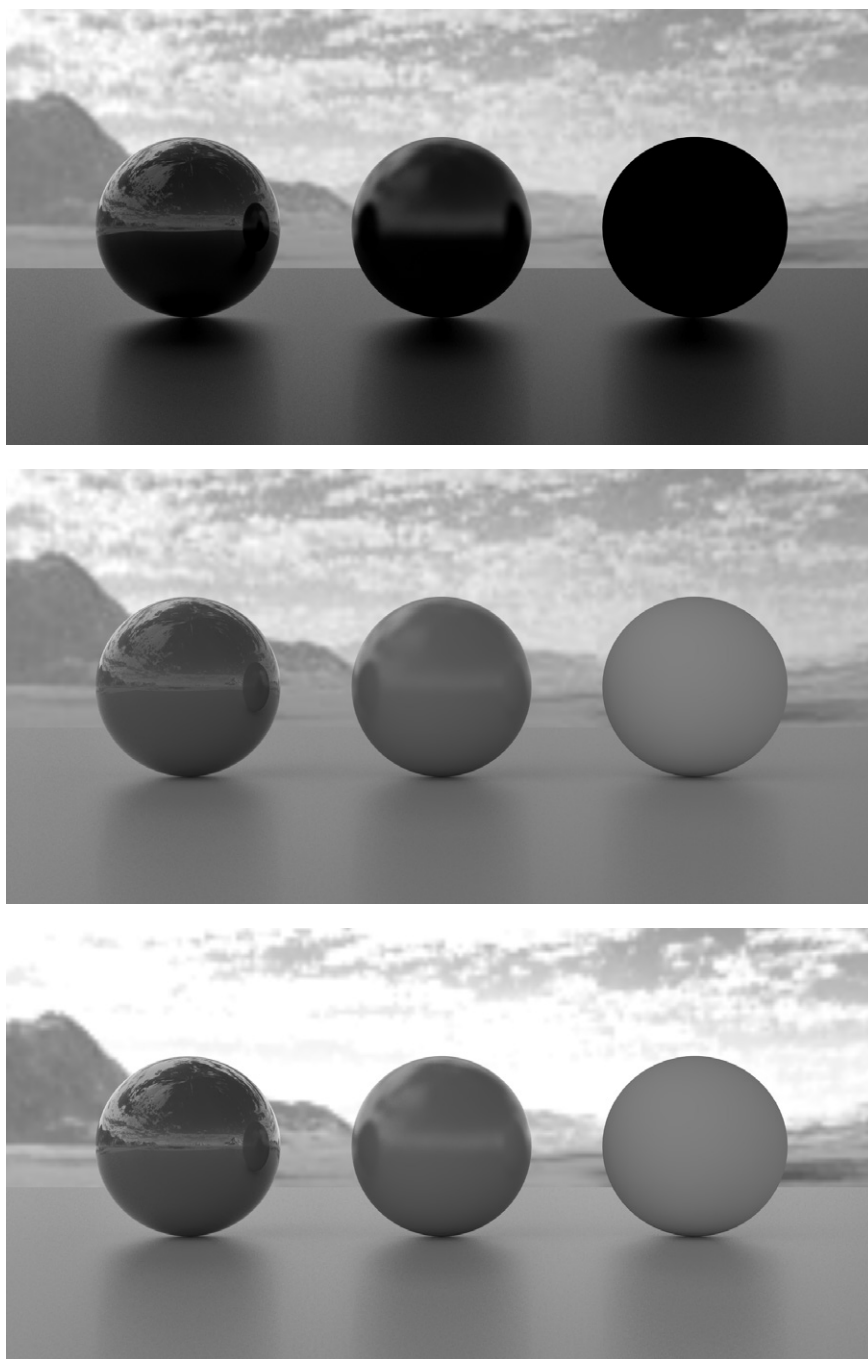


Figura 6.40 IBL. Ejemplo 2. Un mapa. Con FG y Skylight: a) sin skylight, b) con skylight y mapa LDR, c) con skylight y mapa HDR



como fondo en lugar de un mapa virtual pues el cálculo de iluminación avanzada procesará este objeto del mismo modo que cualquier otro objeto de la escena. Lo mismo cabe decir para la simulación de escenas por métodos de iluminación global ligados a materiales con un mapa de entorno. Si el mapa de entorno se aplica como textura a un objeto concreto podremos sacar partido de las propiedades de materiales avanzados. Concretamente, podremos hacer que el material sea autoiluminante con lo que podrá ser procesado por métodos de irradiancia o *final gather*.

El objeto envolvente puede ser una esfera, un hemisferio (*sky dome*) o un cubo (*sky box*). Este último recurso se utiliza principalmente en juegos de vídeo pues aunque da lugar a pequeñas imperfecciones en las aristas estos defectos pasan desapercibidos en una animación. La esfera envolvente se utiliza en casos en los que la vista puede dirigirse en todas direcciones, como ocurriría en una simulación de un vuelo. Como nuestros ejemplos están dirigidos principalmente a aplicaciones arquitectónicas, la mejor opción, que por otro lado es una de las más corrientes, es un hemisferio. El hemisferio puede ser creado como un recurso independiente e insertarlo en la escena cada vez que se necesite. Para ello hacer lo siguiente.

1. Crear un hemisferio que abarque toda la escena prevista (para reutilizarlo bastará con reescalarlo a medida de la escena) y situarlo en 0,0,0.
2. Convertir el hemisferio a malla poligonal. Invertir las normales de sus caras. En *Object Properties* activar *Back Face Cull* para que en el visor solo se muestren las caras con la orientación correcta. De este modo podremos ver el interior desde el exterior y solo ver el interior cuando la cámara esté dentro. Borrar las caras inferiores pues usaremos un plano como base. Si se quiere, aplastarlo ligeramente utilizando *soft selection* o un modificador FFD o un escalado no uniforme en vertical para que se adapte mejor al mapa de fondo.

3. Añadir al hemisferio un mapa UVW. Cambiar su tipo a cilíndrico y hacer que su altura y diámetro se ajusten al hemisferio. También se puede utilizar una proyección esférica pero la cilíndrica suele dar mejores resultados para la vista de fondo porque la deformación es menor. Aunque desde el punto de vista de la iluminación o los reflejos es preferible la esférica por lo que, según los casos convendrá combinarlas.

La disposición general se muestra en la primera figura adjunta de este grupo. Una vez que se cuenta con este objeto envolvente continuar como sigue.

1. Comprobar que el plano de base y las esferas tienen asignados los materiales descritos al comienzo, en la sección “Escenario y mapas...”.
2. Asignar al hemisferio un material avanzado de tipo *Arch&Design*. Hacer que este material sea autoiluminante. Darle un valor inicial de 1.0 cd/m<sup>2</sup> que ajustaremos más adelante. En el grupo *Glow Options* marcar las opciones “Visible in reflections” e “Illuminates the Scene (when using FG)”. Esta última opción es indispensable para que el sistema procese la iluminación.
3. Ajustar la configuración inicial para hacer un comprobación previa. Entrar en *Render Setup*, comprobar que está activado FG con valores bajos. Luego ir al panel de *Environment*, activar el *mr Photographic Exposure Control* y darle un valor de EV 1.0. Esto no es estrictamente necesario pues, en este caso, podríamos prescindir del control de exposición pero nos facilitará las comparaciones con los ejemplos posteriores. *Render* para comprobar que el resultado es básicamente correcto.
4. En el editor de materiales tiene que haber dos mapas, uno JPG y otro EXR, creados con los procedimientos descritos al comienzo, en el apartado “Escenario y mapas utilizados en los ejemplos...”. Arrastrar el JPG sobre el pequeño icono cuadrado que hay junto del color de *Filter* en la sec-





ción de *Self Illumination* del material asignado al hemisferio. Si se quiere, ajustar los valores de *offset* para que se visualicen determinadas zonas del mapa en lugar de otras.

5. *Render*. En función del resultado, modificar el valor de *Physical Units* de autoiluminación del material Hemisferio. En la segunda figura adjunta de este grupo se ha aumentado a 3.0.
6. Aumentar la calidad de salida. En la figura adjunta se ha utilizado calidad de *antialiasing* alta (Min, Max: 1,16) y FG de calidad media con 2 rebotes.
7. Substituir el mapa JPG por el mapa EXR. Si se han modificado valores de *offset* copiarlos.
8. *Render*. Los resultados, principalmente en los reflejos, serán más nítidos. Pero las diferencias no son muy grandes.

### Ejemplo 2. Un mapa. Con FG y *Skylight*

Partimos de misma escena que en el ejemplo anterior pero sin el hemisferio envolvente. En este caso el procedimiento se reduce a asignar un mapa de entorno, crear una luz de tipo *mr sky* ligada al mapa de entorno y ajustar la

configuración de cálculo. También se puede utilizar una luz de tipo *skylight standard* pero la *mr skylight* nos dará más opciones. Para ver mejor las diferencias utilizaremos primero un mapa LDR y luego un mapa HDR, como en el ejemplo anterior.

1. Abrir el Editor de Materiales. Abrir en paralelo el panel de entorno desde el menú *Rendering/Environment*. Seleccionar el mapa LDR y arrastrarlo sobre el botón de mapa de entorno.
2. Hacer que el mapa se muestre en el visor (atajo Alt-B, marcar la opción “use scene environment” y “display background”. Luego, desde la sección *Coordinates*, del *bitmap*, ajustar su posición para que se visualicen las zonas que nos interesan. El procedimiento es el mismo que en el ejemplo anterior y servirán los mismos valores. Cambiar *Blur* a 0.01 (el valor mínimo) para que el fondo quede algo más enfocado.

El valor de *Blur* afecta al enfoque o desenfoque del mapa en función de la distancia (en unidades globales): cuanto más lejos está el mapa mayor es el desenfoque. Se utiliza

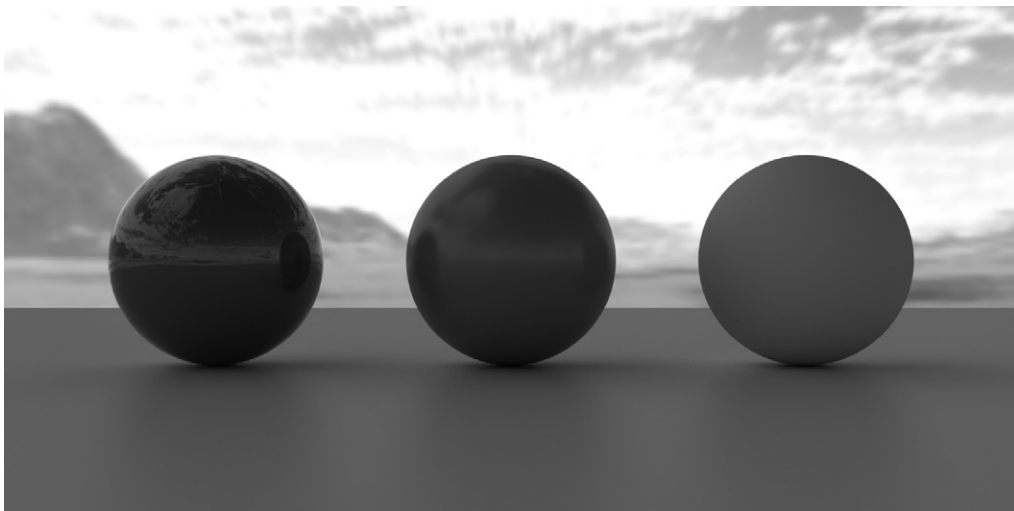


Figura 6.41 IBL. Ejemplo 3. Un mapa. Con iray



sobre todo para prevenir *aliasing* en texturas pero también conviene reducirlo al mínimo en este caso. *Blur Offset* afecta al enfoque o desenfoque del mapa sin tener en cuenta la distancia (en unidades locales). Se utiliza sobre todo para afectar directamente a los detalles de una imagen o textura. Su valor predeterminado (0.0) ya está reducido al mínimo.

3. Crear una luz y apagarla. De este modo se desactivan las luces predeterminadas de 3ds Max y nos aseguramos de que no hay ninguna luz activa. Si ahora activamos el control de exposición *mr photographic* con un valor EV 1.0 y activamos *render* los objetos mates aparecerán de color negro pues FG no procesa los mapas de entorno y, en consecuencia, estos objetos no reciben iluminación. Solo los objetos especulares o semiespeculares serán visibles, en la medida en que reflejen el fondo. Para que reciban luz necesitamos al menos una luz celeste que los ilumine.
4. Crear un sistema de luz diurna. Editarlo y desactivar el sol.
5. Editar el control de exposición y hacer EV=15.0.
7. *Render*.

Ahora si veremos los objetos. Para comprobar la diferencia substituir el mapa LDR por el mapa HDR. Así obtendremos resultados como los de las figuras adjuntas que muestran como el uso del mapa HDR hace que los objetos tengan mayor contraste y las reflexiones sean más ricas de contenido.

El uso de la luz celeste, la *mr Sky* nos permite por añadidura ajustar la intensidad y el color del resultado sin que afecte al fondo. En la sección *mr Sky Parameters* tenemos la posibilidad de variar la intensidad con el multiplicador *y*, en *Advanced Parameters*, tenemos por añadidura la posibilidad de variar el tono (Red/Blue Tint) y la saturación, tal como hemos visto en el capítulo dedicado a este tema.

### Ejemplo 3. Un mapa. Con *iray*

Continuar con la misma escena anterior.

1. Eliminar la *mr Sky*. En la escena solo deberá haber luces apagadas (para que queden desactivadas las predeterminadas por 3dsMax)
2. Desactivar el Control de Exposición.
3. Ir a *Render Setup*. Cambiar el motor de render a *iray*. Ajustar el tamaño de salida y

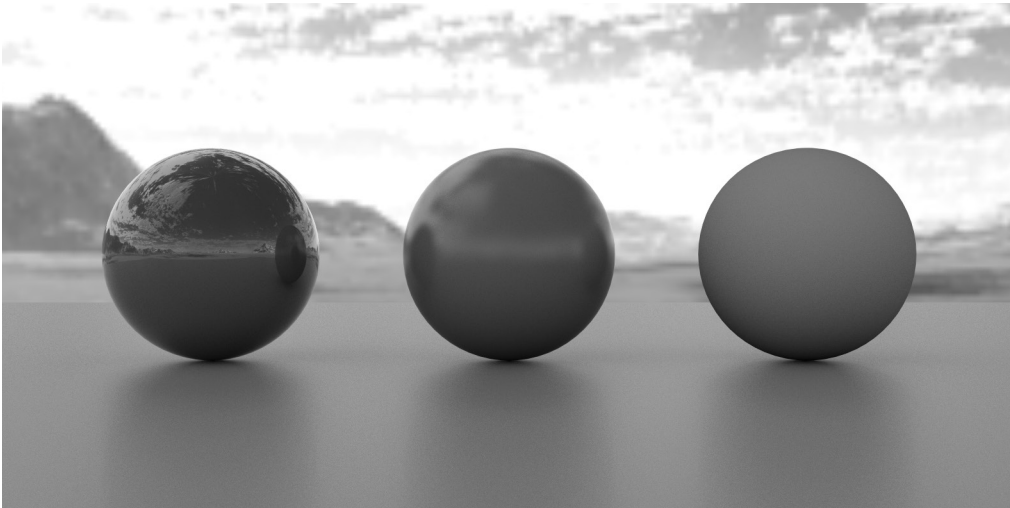


Figura 6.42 IBL. Ejemplo 4. Un mapa. Con IBL nativo



las iteraciones. En la figura del ejemplo se han utilizado 500 iteraciones. Dejar el resto de valores predeterminados (*antialiasing* tipo Gauss).

4. *Render*. El resultado deberá ser similar al de la figura adjunta.

Como ya he dicho la principal ventaja es que es muy sencilla pero también la más limitada. Como puede apreciarse en la figura adjunta el mapa de fondo se procesa bien como reflejo. Puede ajustarse la intensidad de la iluminación aumentando el valor de *Output* en la sección correspondiente de los parámetros del mapa. Pero esto haría que el fondo quedase excesivamente quemado. Este es el principal inconveniente de este método, además de su lentitud: no podemos ajustar por separado el mapa de iluminación y el de fondo.

#### Ejemplo 4. Un mapa. Con IBL nativo

Como he explicado más arriba hay dos posibilidades de utilizar IBL con mental ray: descargar un script que permite usar IBL con las versiones previas a la 2014 (desde las versiones 3.8 de mental ray que aparecieron en 2011) y que incluyen opciones adicionales de incorporación de *importons* a *Final Gather* o de utilizar como alternativa *irradiance particles*. O bien utilizar directamente las nuevas configuraciones de la versión 2014.

Para utilizar la primera hay que activar un *script*. El mejor que he encontrado es el que he mencionado en el apartado “alternativas de configuración”, el desarrollado por Artur Leao. Para utilizarlo, arrastrarlo dentro del programa (es posible que haya que reiniciar). Luego ir a menú *Customize* y buscar, en la lista de *Actions*, “mr manager”. Arrastrarlo a un menú o una *toolbar* en donde quedará disponible con el nombre “mrOM” y todo lo que habrá que hacer es activarlo. Al activarlo aparecerá un panel con cuatro secciones: a) *Unified Sampling*, presenta el nuevo método de *Antialiasing* que he descrito en el capítulo 4 y que simplifica la configuración y que es también el mismo que se utiliza en la versión

2014; b) *Progressive Rendering*, que también se ha descrito en el capítulo 4 y da mejores y más controlables resultados. Veremos más abajo que también hay un modo de activarlo en la versión 2014; c) *IBL*. Se puede activar esta opción de tres modos (*light*, que es el predeterminado y el que da mejores resultados, *approximate* y *automatic*). Incluye otros parámetros que no detallo; d) *Importons*, que incluye unos pocos parámetros, el principal de los cuales sería la densidad, e) *Irradiance Particles*, que incluye parámetros característicos de este recurso, que tampoco detallo pero que son muy similares a los de *Final Gather*.

No lo describo con detalle porque no tengo garantías ni sobre ni su continuidad ni sobre su eficacia en relación con otros métodos. Pero lo cito porque puede haber lectores interesados en probarlo y, principalmente, para dar indicaciones sobre la diversidad de métodos que se están ensayando y probando para que los recursos teóricos y técnicos que hemos visto en capítulos anteriores puedan llegar a utilizarse.

#### § § §

Para IBL con la versión 2014 las cosas son bastante más sencillas. El procedimiento básico es el mismo que en el ejemplo 2 (un mapa, FG con *Skylight*) aunque en este caso utilizaré solo un tipo de mapa HDR. Se puede comprobar que con un mapa de tipo LDR los resultados son similares pero con algo menos de calidad.

1. Proceder inicialmente como en el ejemplo 2, es decir, desde el Editor de Materiales, crear un mapa (de tipo EXR) y llevarlo al panel de *Environment* para asignarlo como mapa de entorno. Luego crear un sistema de luz diurna y desactivar el sol (también se puede utilizar una *Skylight Standard* pero los resultados son algo peores y menos controlables).
2. Ir a *Render Setup/ Global Illumination/ Skylights & Environment Lighting (IBL)*. En el grupo *Skylight Mode* hay dos

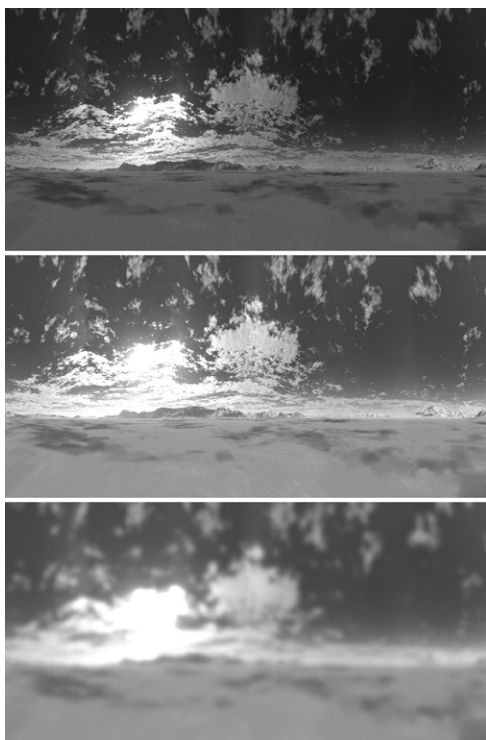
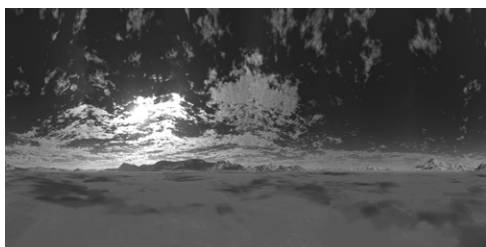


Figura 6.43 Ejemplo 5. Tres mapas: a) Mapas utilizados para a) el fondo (zzz\_Bkg.jpg), b) los reflejos (zzz\_Rfl.jpg), c) el entorno (zzz\_Env.exr).



BKG (ldr)

RFL(hdr)

ENV (hdr)

Figura 6.44 Ejemplo 5. Tres mapas. Los mismos mapas mostrando su tamaño relativo

opciones: "Skylight illumination from FG" y "Skylight illumination from IBL".

3. Si se escoge la primera el resultado será como el que hemos visto en el ejemplo 2.
4. Escoger por tanto la segunda. Al hacerlo quedarán disponibles dos parámetros: a) *Shadow Quality* con un valor de 0.5 predeterminado, sobre un rango de 0.0 a 10.0), b) *Shadow Mode*, predeterminado a "Transparent", más preciso pero más lento. La otra opción es "Opaque", menos preciso pero más rápido. Dejar los valores predeterminados (o reducir la calidad) para pruebas iniciales a baja resolución pero aumentar *Quality* a 2.0 para la imagen final.
5. Desactivar FG.
6. *Render*. El resultado final deberá ser como el de la figura adjunta. La calidad es mayor y el tiempo de cálculo más rápido que con FG.

Aunque en esta escena no tiene demasiado sentido, en otros casos puede ser interesante utilizar *Progressive Render*. Y, como veremos en la siguiente sección, también puede ser interesante combinar IBL con *Ambient Occlusion*. En ambos casos habrá que utilizar un nuevo recurso de mental ray que permite añadir estas y otras técnicas: *String Options*, que se encuentra en *Render Setup/ Renderer*. Para ver como funciona *Progressive Render* ir a la sección correspondiente del capítulo 4. Y para ver como funcionan las *String Options* con *Ambient Occlusion* consultar la ayuda del programa.

#### Mapas utilizados en el último ejemplo que sigue

Como debe resultar claro por las figuras adjuntas a los ejemplos anteriores, el mapa utilizado para la iluminación no funciona muy bien como mapa de fondo. Para que la calidad fuera equiparable a la de las formas principales deberíamos aumentar la resolución hasta unos límites poco prácticos. Y, por añadidura, esto podría complicar la iluminación pues lo que necesitamos para el mapa de iluminación



es más bien lo contrario, como ya he indicado: un archivo sin una resolución excesiva y sin zonas de alta frecuencia (líneas finas, bordes, saltos bruscos de valores). Por ello, es preferible utilizar dos variantes del mismo mapa: una como mapa de fondo pero aplicado en modo Pantalla (o en modo Cilíndrico o Esférico con repetición) con buena resolución y contraste y otra como mapa de iluminación, con menos resolución y desenfocado para eliminar zonas de alta frecuencia.

Por otro lado, el mapa utilizado tampoco es lo ideal para los reflejos. Pues, por un lado, no necesitamos tanta resolución y, por otro lado, no tiene por que coincidir con la imagen de fondo sino más lo contrario pues se supone que está reflejando zonas opuestas. Y, además, las características adecuadas para los reflejos no son las mismas que para los fondos.

Resumiendo:

a) Para la iluminación global necesitamos un mapa HDR que incorpore valores lumínicos en un rango dinámico alto. Pero no es necesario que tenga mucha resolución. Y, además, no solo no es necesario que tenga mucha definición (zonas de alta frecuencia que representen bordes, detalles de la escena) sino que esto resultará perjudicial pues se traducirá en manchas y ruido en la imagen de salida.

Concretando. Para una imagen de salida que esté entre los 1024 y los 2048 píxels de anchura nos bastaría con menos de la mitad de esta resolución de salida, pongamos que entre 512 y 1024 píxels. Y debe estar suavizada para eliminar zonas de altas frecuencias. Esto quiere decir que deberemos hacer una copia de nuestro mapa original, llevarla a Photoshop, reducir su tamaño y aplicarle un filtro de desenfoque gaussiano dándole un nombre adecuado tal como “z\_Env.exr”, en donde “z” sería el nombre del original. Me referiré a este mapa como “ENV”.

b) Para el fondo necesitamos un mapa LDR de alta resolución. Debe ser LDR pues no necesitamos dedicar memoria a almacenar valores que no interesan para una imagen

que no va a iluminar nada. Y debe ser de alta resolución porque va a ser una imagen visible y porque debe abarcar el conjunto de la escena, más aún si la escena va a ser animada.

Concretando. Para una imagen de salida de las dimensiones citadas tendremos que utilizar el archivo original, que debería tener del orden de 8 veces la anchura de salida si va a ser utilizado en múltiples vistas. Alternativamente, podemos utilizar la misma dimensión de salida o algo mayor, aunque esto nos obligará a ajustar la vista para cada toma. Una solución intermedia es hacer que tenga el doble, lo que cubriría cerca de la mitad de la escena. El archivo original debe, en cualquier caso, ser traducido a LDR. Esto quiere decir que deberemos hacer una copia de nuestro mapa original, llevarla a Photoshop o a *Picturenaut* o a otro programa similar, aplicarle un algoritmo adecuado de *Tone Mapping* (más claro o más oscuro o con más o menos contraste, según nos interese) y guardarla en formato JPEG dándole un nombre adecuado tal como “z\_Bkg.jpg2”, en donde “z” sería el nombre del original. Me referiré a este mapa como “BKG”.

c) Para los reflejos necesitamos un mapa LDR con buena resolución o un mapa HDR que incorpore valores lumínicos en un rango dinámico alto pues en los reflejos se da un alto grado de contraste entre luces o zonas muy iluminadas y zonas menos iluminadas. En el primer caso necesitamos una resolución intermedia y en el segundo, como solo será visible en las partes reflectantes de los objetos, no es necesario que tenga demasiada resolución.

Concretando. Para una imagen de salida de las dimensiones citadas nos bastará con una dimensión del orden de menos de la mitad del mapa original si bien esto depende de cada caso. Esto quiere decir que deberemos hacer una copia de nuestro mapa original, llevarla a Photoshop, reducir su tamaño y guardarla con un nombre adecuado tal como “z\_Rfl.exr”, en donde “z” sería el nombre del original. Me referiré a este mapa como “RFL”.





Resumiendo todo lo anterior con un ejemplo. Supongamos que queremos una resolución de salida de 1600 px de ancho. En este caso necesitaremos:

Un mapa de fondo, **BKG**, de una resolución al menos 4 veces la de salida (el doble si queremos cubrir 360°), es decir de 6.400 px de ancho, y de tipo LDR.

Un mapa de entorno, **ENV**, de una resolución de poco más de la cuarta parte de la de salida, pongamos que 512 px de ancho para este ejemplo, de tipo HDR y editado para que de desenfocado, eliminando las zonas de alta frecuencia.

Un mapa para los reflejos, **RFL**, con una resolución algo superior a la de salida, pongamos que 2048 px de ancho para este ejemplo, de tipo HDR y que preserve los detalles del original.

### Ejemplo 5. Dos o Tres mapas. Iluminación, fondo, reflejos

Parto por tanto de la base de que, tal como he descrito en el apartado anterior, ya contamos con tres mapas: BKG, RFL y ENV, (*z\_Bkg.jpg*, *z\_Rfl.exr*, *z\_Env.exr*, en donde “z” sería el nombre del archivo original). Bastaría utilizar dos mapas en muchos casos en los que no hay reflejos o son poco visibles. Pero para generalizar, partiré de la base de que queremos controlar por separado la escena por medio de tres mapas. Adaptarlo a un caso en el que bastarían dos mapas resultará muy sencillo pues todo lo que hay que hacer es anular uno de los mapas.

El método de cálculo puede ser variable. Pero, para que la descripción sea más general, utilizaré la segunda variante de las anteriores, *FG* con una *skylight*, que da resultados algo inferiores de calidad pero puede aplicarse a muchos tipos de programas y versiones con relativa facilidad. El mismo procedimiento básico puede adaptarse a las otras alternativas que hemos visto. Partimos de la misma escena que en los ejemplos anteriores. A partir de ahí, proceder como sigue, repitiendo el proceso del ejemplo 2.

En principio hay dos modos de aplicar estos dos mapas. El primero, más sencillo de utilizar pero menos general y que no da mejores resultados, es utilizar como mapa de entorno un *shader* especial de *mental ray* denominado *Color Override/ Ray Type Switcher*. Este *shader* permite asignar diferentes mapas a diferentes canales de procesamiento, concretamente a *Eye Ray* (cámara), *Transparency*, *Reflection*, *Refraction*, *Shadows*, *Final Gather*, *Photons* y *Environment*. En nuestro caso bastaría con asignar el mapa de fondo (BKG) a *Eye Ray*, el mapa de reflejos (RFL) a *Reflections* y el mapa de entorno (ENV) a *Final Gather*.

El segundo, más general y que da resultados algo mejores, es utilizar como mapa de entorno otro *shader* especial de *mental ray* denominado *Environment/ Background Switcher* que en realidad puede considerarse como una variante simplificada del anterior. Este *shader* solo tiene dos entradas: para insertar un mapa de fondo y para insertar un mapa que procese el entorno en la medida en que afecta a los reflejos. Como necesitamos un tercer mapa, del que se derivará la iluminación global, podemos asignar el mapa de entorno a una *skylight*.

El procedimiento a seguir sería el siguiente:

- 1 Desde el Editor de Material crear tres mapas en otros tantos visores para cada uno de los tres mapas BKG, RFL y ENV que vamos a utilizar. Editarlos y, en *Coordinates*, hacer que se apliquen como mapa de entorno en lugar de textura (escoger la opción “environment map”) y que su proyección sea de tipo *Screen* en el primer caso y *Spherical* en los otros dos. Asignar provisionalmente el primero, BKG, al entorno, y modificar sus coordenadas (*UV Offset* y *UV Tiling*) para que el tamaño y la posición sean los que interesen. También se puede seleccionar una determinada zona si interesa hacerlo: para ello, en la sección *Bitmap Parameters*, en el grupo *Cropping/ Placement*, presionar el botón *View Image*.





Arrastrar las asas de la ventana de recorte para que se adapten a la zona que nos interese. Luego cerrar esta ventana y marcar las opciones *Crop* y *Apply*.

- 2 Crear un cuarto mapa de tipo *Environment/Background Switcher*. Este mapa, como ya he dicho, solo tiene dos entradas, una para el fondo (*Background*) y otra (*Environment/Reflections*) para los reflejos de entorno. Arrastrar BKG al primero y RFL al segundo.

Luego arrastar este mapa, *Environment/Background Switcher*, como instancia, sobre el botón de mapa del panel de *Environment* para que el motor de *render* lo procese.

- 3 Crear una luz estándar de tipo *skylight*. Marcar la opción *Sky color* y *Map* y arrastar el mapa ENV sobre el botón que hay junto a *Map* de esta luz. Aumentar algo el multiplicador de la luz en función de los

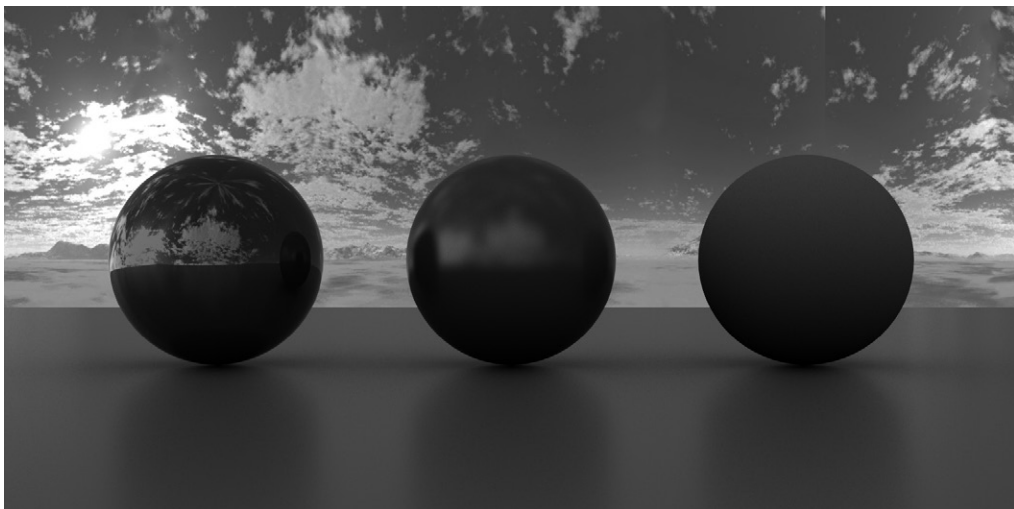
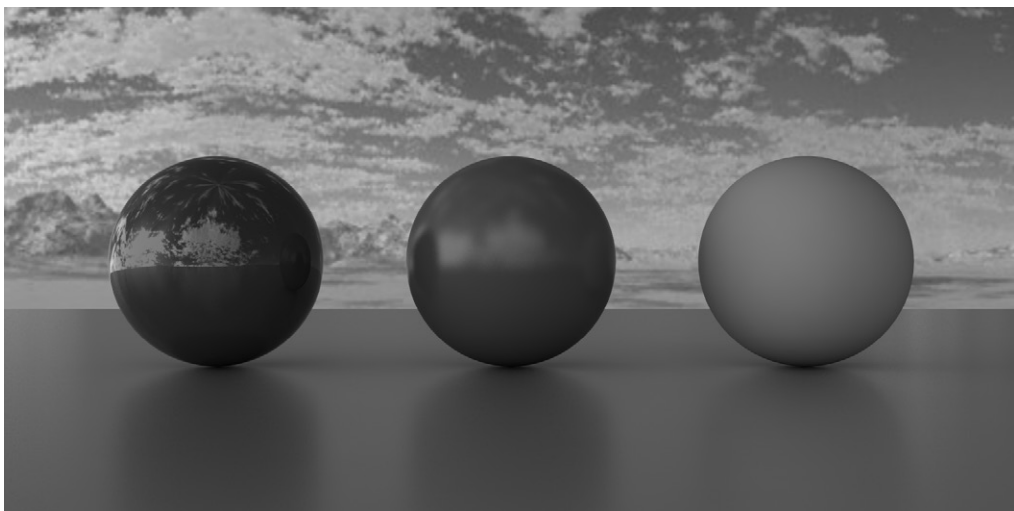


Figura 6.45 IBL. Ejemplo 5. Tres mapas:

a) Con Final Gather y una skylight standard, b) Sin Final Gather y con IBL nativo y una skylight standard



resultados. En la figura adjunta se ha aumentado a 1.5.

- 4 Desactivar el Control de Exposición (pues estamos utilizando luces simples).
- 5 Ir a *Render Setup* y activar FG con valores bajos para pruebas previas y valores más altos al final.
- 6 Si queremos que los reflejos sean más intensos se puede aumentar el valor de salida desde la sección *Output* del mapa RFL. En la figura adjunta se ha elevado algo.

El mismo procedimiento se puede seguir utilizando con el IBL nativo de la versión 2014 de 3ds Max (3.11 de mental ray). En este caso, como ya hemos visto, hay que ir a *Render Setup/ Global Illumination/ Skylights & Environment Lighting (IBL)*, seleccionar la opción "Skylight illumination from IBL" y desactivar *Final Gather*.

Si analizamos los resultados veremos dos diferencias importantes con respecto a los anteriores: la distribución de la iluminación, más apreciable en el plano de base que tiene una ligera reflectancia, es más uniforme lo que, en general, será preferible. Y el reflejo en la bola tiene menos detalle pues hemos reducido un poco la resolución del mapa utilizado. Pero estas diferencias más bien favorecen el resultado y, sobre todo, aceleran considerablemente el cálculo.

### Smart IBL (sIBL)

Si el procedimiento de asignar tres mapas a una escena se va a repetir muchas veces nos encontraremos con un proceso muy repetitivo pues, a partir de un mapa HDR/EXR original habrá que generar otros dos siguiendo criterios muy similares. Hay una utilidad gratuita, desarrollada por Christian Bloch, autor de un buen libro sobre HDR (ver referencias), en colaboración con Christian Bauer, Volker Heisterberg y Chris Huf, que simplifica este proceso. La utilidad se denomina sIBL (*smart IBL*), funciona con los principales programas de simulación, Maya, 3ds Max, SoftImage, V-Ray, Blender, Cinema 4D, etc. Puede descar-

garse (en 2014) de <http://www.hdrilabs.com/sibl/index.html> o, si hubiera cambiado, haciendo una búsqueda con estas palabras clave.

### Uso de IBL con otras aplicaciones

Como en casos anteriores solo incluyo el *software* más utilizado en arquitectura y diseño, dejando a un lado otros motores de primera fila pero que se utilizan más en estudios muy especializados.

El procedimiento básico es muy similar al que hemos visto. Doy solo algunas indicaciones generales sin entrar en el detalle de los parámetros concretos que se utilizan en cada caso.

Con las últimas versiones de **V-Ray** la luz de tipo *dome* (equivalente a la *skylight* de *mental ray*) ha ampliado sus propiedades para aceptar todo tipo de mapas que determinan la intensidad de luz en cada dirección del hemisferio abarcado por la *dome light*. El programa utiliza *importance sampling* para distribuir los rayos de modo que se envíen más en las direcciones con mayor intensidad luminica. La *dome light* puede rotarse, lo que se traduce en diferentes distribuciones de la iluminación. También puede complementarse el cálculo con *photon mapping* para obtener efectos cáusticos.

Con **Maxwell** se pueden ajustar de diferentes modos cuatro canales: *illumination*, *background*, *reflection* y *refraction*, modificando en cada caso el mapa, la intensidad y la orientación. Y también los tipos: *dome*, *physical*, *hdr*, de cada canal. Los mapas pueden ser de tipo HDR, EXR o MXI (Maxwell).

Con **Blender** se puede utilizar IBL por medio de *Blender Cycles* y de *Yafaray*. La versión 2.68 de *Cycles* soporta dos tipos de *IBL rendering*: con proyección *Equirectangular* y con proyección *Mirror Ball*. También pueden utilizarse estos mismos tipos con *Yafaray* (que utiliza los términos equivalentes *Lat/Long* o *Spherical* para *Equirectangular* y *Angular Mapping* para *Mirror Ball*). Se utiliza por medio de mapas del tipo citado asignados a *Environment*.



Con **Cinema 4D** también se puede utilizar IBL. Un modo de hacerlo es creando un hemisferio que rodee la escena, aplicándole un material, asignando un mapa HDR al canal de luminancia del material y utilizando radiosity como método de cálculo. Otro modo más eficiente es por medio de *irradiance caching* asignando igualmente un material adecuado al envolvente. También puede utilizarse el recurso que he citado más arriba, sIBL (*smart IBL*).

## 6.8 Oclusión ambiental (*Ambient Occlusion*)

### *Ambient Occlusion* como sistema de cálculo de iluminación sin luces

Hay que comenzar por recordar algo que ya he avanzado en el capítulo 3 pero que en el contexto de una aplicación práctica se entenderá mejor. La oclusión ambiental, *Ambient Occlusion*, AO, no es un sistema de cálculo de la iluminación pues no tiene en cuenta las luces de la escena y, de hecho, se utiliza al margen de ellas pues podemos obtener resultados de interés sin utilizar ninguna luz virtual.

Por otro lado, como también hemos visto en el capítulo 4, las GPU recientes incorporan sistemas de cálculo automático de la oclusión ambiental por lo que, en muchos casos, se utiliza como un recurso independiente que mejora la calidad de la escena, no solo por lo que respecta al cálculo final sino incluso como recurso interactivo de visualización. En muchos programas que están optimizados para su uso con GPUs relativamente avanzadas, puede activarse esta opción para que lo que se muestra en pantalla incorpore, si interesa, sombreados difusos calculados a partir de la oclusión ambiental.

De todos modos los mejores resultados los obtendremos si podemos acceder a los parámetros que están en juego con este método. Por esta razón, la parte principal de esta sección estará dedicada a describir el funcionamiento de *Ambient Occlusion* como *shader*

independiente pues, además de que amplía sus posibilidades de utilización, permitirá entender mejor el funcionamiento interno de este recurso.

### Descripción general

Los principios teóricos de los métodos de cálculo de la oclusión ambiental se han descrito en el capítulo 3. En esta sección se describen técnicas específicas de uso. Esta técnica no puede utilizarse con un motor de *render* de tipo *scanline* pues requiere un motor basado en trazado de rayos. Describiré su uso solamente con *mental ray*. Pero este uso es relativamente sencillo y las explicaciones que siguen se podrán adaptar fácilmente a otros programas.

Hay dos modos de llevarla a cabo, como también se ha visto en el capítulo 3, a través de asignaciones a materiales o a través de representaciones independientes (*render passes* o *render elements* en 3ds Max). Requiere, por tanto, o bien materiales que integren estas capacidades, como es el caso del material *Arch&Design*, o bien que permitan incorporarlas, como es el caso del material *mental ray*, o bien sistemas de cálculo que permitan hacer un pase independiente de este tipo, como también es el caso si se utiliza *mental ray*. Si se utiliza este último método, los pases independientes, hay que componer el resultado final utilizando métodos de post-producción. Los resultados suelen ser mejores y más controlables si se utiliza así. Pero también es menos directo y muchos usuarios preferirán obtener un resultado directo aunque sea menos perfecto que tener que pasar por el proceso de grabar dos salidas y componerlas con un programa adicional.

### Parámetros

Como ya he dicho, se puede usar de dos modos: a través de un material que incluya parámetros básicos de Arch&Design o bien a través de un *shader* específico con una serie más completa de parámetros. Doy la descrip-



ción de los parámetros generales del *shader* específico pues en el primer caso solo estarán disponibles dos de estos parámetros, *Samples* y *Max Distance*, además de unos controles de color adicionales que ya describiré en el ejemplo correspondiente a este caso. Para explicaciones más diversificadas ir a la Ayuda del programa que se esté utilizando.

Para utilizar el *shader* específico, la mejor opción es partir de un material genérico, *mental ray*, y en la sección *Basic Shaders*, pulsar el botón *Surface* y escoger como mapa (*shader*), *Ambient/ Reflective Occlusion*. También se puede utilizar un material *Standard* o *Arch&Design* y desplazarse a la sección *mental ray connection* en donde aparecerán los mismos botones de configuración que con el material *mental ray* y, como antes, presionar el botón junto a *Surface* para acceder al *shader*. Los parámetros que aparecerán disponibles en cualquier caso, son los siguientes (doy los valores predeterminados entre paréntesis).

**Samples** (16). Especifica el número de rayos que se envían desde cada muestra para analizar la oclusión. Cuantos más rayos se emitan más suave es el resultado y mayor el tiempo de cálculo.

**Spread** (0.75). Especifica la apertura de los rayos, el ángulo de la zona cónica que computa la oclusión. Con 0.0 sólo se computa un punto. Con 1.0 se computa todo el hemisferio en torno al punto. El rango va de 0.0 a 1.0 (aunque se pueden especificar valores mayores pero serán irrelevantes). Con valores pequeños el efecto de oclusión será más localizado. Con valores altos resultará más suave y también necesitará más muestras.

**Bright** (blanco), **Dark** (negro). Especifican el color del mapa en donde no hay oclusión (blanco) o hay plena oclusión (negro). Si se escogen grises claros y oscuros el resultado será similar pero más atenuado, con menor contraste.

**Max distance**. Especifica el radio dentro del cual el algoritmo analiza si hay objetos que ocluyen el punto considerado. Valores pequeños restringen el efecto (sólo aparecerán pequeñas zonas de sombra en algunos

rincones) pero se procesan más rápido. Valores grandes cubren áreas mayores (aparecerán sombras más amplias) pero son más lentos. El valor predeterminado no es muy fiable pues debe depender de las unidades de la escena. Una referencia adecuada es utilizar la longitud de un elemento que resalte notoriamente sobre el plano analizado. Por ejemplo, si se quiere aplicar a una fachada con balcones de 100 cm de profundidad, este mismo valor es un buen punto de partida.

**Reflective** (desactivado). Si se activa, las muestras se toman alrededor de la dirección de reflexión. Esto genera efectos de *oclusión reflectiva* que pueden extender notoriamente el efecto de la reflexión.

**Type (occlusion)**. Incluye tres tipos: *Occlusion* (predeterminado) *Environment* y *Bent Normals*. Este parámetro especifica el tipo de cálculo. El tipo predeterminado, *Occlusion*, hace que se tome en cuenta la dirección de muestreo de los rayos. Si se escoge *Bent Normal*, se computan las direcciones dadas por las normales. Si se escoge *Environment* se toma en cuenta el mapa de entorno cuyos valores se multiplican por el color dado por el parámetro *Bright* para obtener el valor final.

**Falloff** (1.0). Define la cantidad de atenuación del rayo. Si se aumenta el valor, aumenta la atenuación.

**Incl./Excl. Object ID** (0). Se utiliza para incluir o excluir objetos en la oclusión. Con el valor predeterminado, 0, no se excluye ningún objeto. Un valor positivo (1, 2, etc) indica que solo se tendrán en cuenta para la oclusión los objetos con el índice especificado. Un valor negativo, que los objetos especificados se excluirán de la oclusión. Si por alguna razón se quiere utilizar este recurso habrá que modificar, en las propiedades del objeto, estos índices.

**Non-Self-Occluding Object ID** (0). Funciona de modo similar al anterior pero indicando en este caso la autoclusión. Al igual que el anterior es raramente utilizado.

En la práctica, los parámetros más importantes son *Max distance* y *Spread* y a menudo bastará con ajustar estos dos parámetros.



También convendrá comenzar por valores bajos para *Samples* (16 o menos, por ejemplo 8) para hacer pruebas y luego aumentar estos valores para la salida final (64 o más, por ejemplo 128).

### Ejemplo

Crear una escena, como la de la figura adjunta, que incluya rincones, salientes, etc, en los que se crearían zonas más oscuras por efecto de la oclusión ambiental.

1. Desactivar todas las luces, si las hubiera. Desactivar el control de exposición. Desactivar FG.
2. Desde el *Render Setup/ Processing/ Translator Options*, en el grupo *Material Override*, escoger como material un material *Arch&Design* o un material *Standard* y desplazarse a la sección *mental ray Connection*. O bien, como ya he dicho más arriba, escoger directamente un material de tipo *mental ray* que mostrará directamente esta sección. En cualquier caso, activar la casilla *Enable* del *Material Override*. Así, este material se superpondrá a todos los materiales de la escena.
2. Arrastrar el material al Editor de Materiales para editarlo. En *Basic Shaders / Surface*, pulsar el botón junto a *Surface* (si se usa a través de un material *Standard* o *Arch&Design* habrá que desbloquear previamente este botón) y escoger un mapa de tipo *Ambient/Reflective Occlusion*. Editar este mapa.
3. Sus parámetros ya se han descrito más arriba. El primero que nos importa es *Max Distance*. Los valores adecuados, como ya he dicho en el capítulo 3 y en los apartados anteriores, dependerán de cada caso. En la escena ilustrada, las unidades están en cm y la altura del muro visible es de unos 400 cm. Los valores adecuados estarán en torno al valor de los elementos en relieve. Pero la figura adjunta muestra diferentes resultados obtenidos variando este parámetro.

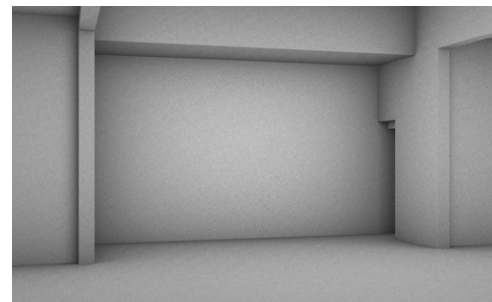
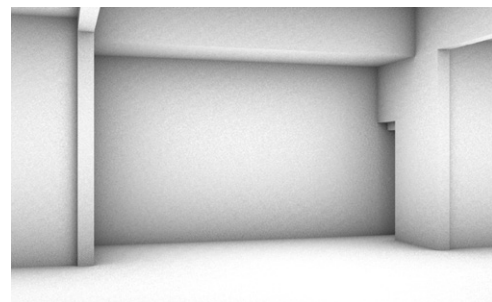


Figura 6.46 Ambient Occlusion. Parámetros predeterminados y valores variables de *Max Distance* (cm): a) 10, b) 50, c) 100, d) 1000.





4. El siguiente parámetro que puede convenir ajustar, aunque no siempre, pues el valor predeterminado suele ser suficiente, es el valor de *Spread*. También se muestran las diferencias en la figura adjunta.
5. Por último, tendremos que ajustar el valor de *Samples*, más aún si hemos aumentado el valor de *Spread* pues el ruido será más visible. También se muestran las diferencias en la siguiente figura.

Así obtenemos una imagen que muestra la distribución de la luz debida exclusivamente a la oclusión ambiental.

Pero esta imagen no muestra ni los materiales, ni la distribución de la luz debida a la iluminación de la escena. Por consiguiente no podemos quedarnos aquí excepto en casos especiales en que nos baste con una representación semejante a la que obtendríamos fotografiando una maqueta. Cuando se pro-

cesa una escena de este modo necesitamos, casi siempre, componer el resultado con otra representación que muestra la iluminación básica y los materiales.

Por tanto necesitamos otra imagen que incluya información sobre los materiales. Dado que la “iluminación” va a correr a cargo exclusivamente de *Ambient Occlusion* nos vendría contar con otra imagen que mostrara información tan solo sobre los materiales. Y a

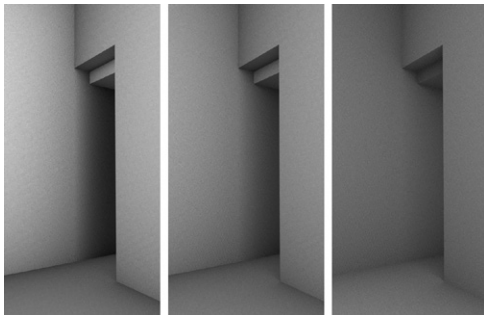


Figura 6.47 Ambient Occlusion. Variaciones con Max Distance 1000 y distintos valores de Spread: a) 0.50, b) 0.75, c) 1.50

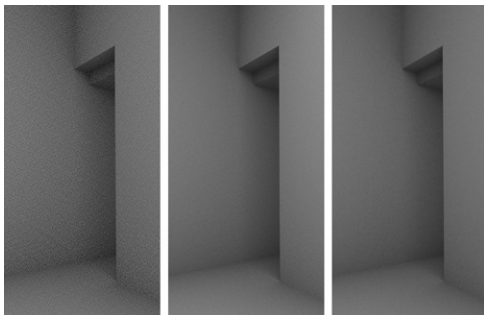


Figura 6.48 Ambient Occlusion. Variaciones con Max Distance 1000, Spread 1.50 y distintos valores de Samples: a) 4, b) 16, c) 64

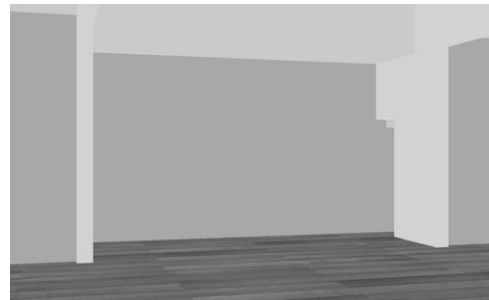
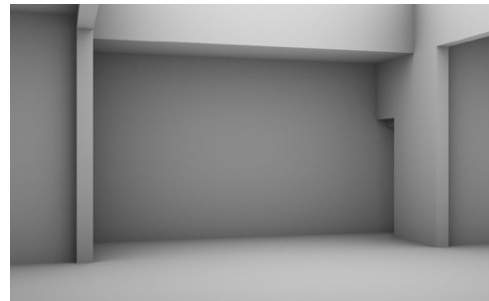


Figura 6.49 Ambient Occlusion. Composición de imágenes: a) Imagen obtenida con Max Distance 1000, Spread 1.5 y Samples 256, b) Imagen plana con información exclusiva sobre colores y texturas de la escena, obtenida mediante el recurso Render Elements, c) Resultado final obtenido combinando las dos imágenes anteriores (original en color)





la que superpondremos la información de la otra imagen.

Esto puede hacerse desactivando el *material override* para generar otra imagen con los materiales de la escena. Pero tendríamos que ajustar la iluminación para que no interfiriera con la de Ambient Occlusion. Será mejor utilizar un recurso, *Render Elements*, que se utiliza para componer imágenes que incluyan sólo determinadas propiedades. Para probarlo, continuar como sigue.

1. Ir a *Render Setup/Render Elements*. Presionar el botón *Add*. Escoger el elemento *mr A&D Level Diffuse*. Asegurarse de que está activado “Enable” tanto para este elemento como para el conjunto (“Elements Active”).
2. Aumentar los valores de *antialiasing* (en el ejemplo se han utilizado un *Min,Max* de 4,16) y cambiar el filtro a Mitchell para mejorar la calidad de las texturas en el caso de que estas incluyen detalles finos o líneas inclinadas.
3. *Render*. Se crearán dos imágenes, una normal (que resultará completamente negra pues no hay luces en la escena) y otra, como la de la figura adjunta, que mostrará tan solo los materiales mates con sus colores o texturas. Guardar esta imagen para combinarla con la que hemos obtenido previamente.

Podríamos haber utilizado este mismo recurso para incluir una imagen que retuviera tan solo la información sobre *Ambient Occlusion*. Para hacer esto, desde *Render Elements*, habría que presionar el botón *Add* de nuevo y escoger *mr Shader Element*. Este *shader* es una especie de comodín que permite escoger cualquier tipo *shader* y utilizarlo para crear un pase adicional en el render. Una vez añadido, en la parte inferior aparecen los parámetros correspondientes y entre un botón que da acceso al navegador de mapa lo que nos permitiría escoger *Ambient/Reflective/Occlusion*. Y, luego, desde el botón de *Render Elements*, arrastrarlo al Editor de Materiales para edi-

tarlo y ajustar sus parámetros. Una vez configurado, al hacer un *render*, aparecería una imagen semejante a la que hemos obtenido al comienzo. Pero los resultados, por lo que he comprobado (no en las últimas versiones) son de peor calidad que los que se obtienen con el procedimiento general que he descrito más arriba.

También podríamos añadir otras imágenes que retuviesen información sobre los reflejos si fuera necesario. O sobre elementos. Pero para simplificar la exposición me limitaré al caso de una escena con objetos mate.

Para componer los dos resultados necesitamos utilizar otro programa de edición de imágenes. El procedimiento, en cualquier caso, es muy sencillo.

1. Desde Gimp o Photoshop, abrir la última imagen (materiales planos) y copiar la anterior (*ambient occlusion*) en otra capa superpuesta. Supondré que la primera capa se ha renombrado como “Color difuso” y la segunda como “AO”.
2. Cambiar el modo de la capa superpuesta, la capa “AO” a Multiplicar. De este modo se combina la información de las dos capas.
3. Es probable que el resultado sea demasiado oscuro. Ajustar las curvas o los niveles para aumentar algo la luminosidad de los valores medios. Luego combinar las capas. El resultado final deberá ser similar al de la última figura adjunta de este grupo.

## 6.9 Análisis y medición de la iluminación

Con 3ds Max se pueden analizar los resultados de una simulación de la iluminación de un modo lo suficientemente exacto como para utilizarlo como recurso en el diseño arquitectónico.

Hay dos modos de hacerlo, uno, más antiguo, requiere utilizar radiosidad y es bastante sencillo pero algo más limitado, además de requerir el uso de un método de cálculo que,



como he dicho más arriba, cada vez se utiliza menos en la práctica real debido a sus inconvenientes prácticos. El otro método, más reciente, funciona exclusivamente con *mental ray* y *Final Gather*. También debe tenerse en cuenta que solo está incluido en la versión *3ds Max Design* (una versión similar en todo al *3ds Max* corriente pero con algunos recursos adicionales dirigidos a arquitectos y diseñadores).

En el siguiente apartado daré un resumen muy breve del uso de radiosisidad y en el siguiente un resumen más extenso y referido a un ejemplo concreto, del uso del método basado en *mental ray*.

## Análisis de la iluminación con radiosisidad

### Recursos

Las mediciones, cuando se usa radiosisidad, son muy sencillas y bastante fiables. Una vez que se ha llevado a cabo un cálculo de radiosisidad, el cuadro de diálogo **Análisis de iluminación** permite obtener información sobre la luminancia, iluminancia, reflectancia y transmitancia de un punto determinado de la escena o sobre los valores medios relativos a un objeto.

Para ello, ir a Menú *Rendering/ Advanced Lighting/ Lighting Analysis*. Se abre un cuadro de diálogo en el que se recoge la información sobre cualquier punto de la imagen en  $\text{cd/m}^2$  (luminancias) o luxes (iluminancias). Mientras el cuadro permanece abierto el cursor se transforma en un cuentagotas (al rebasar el panel). Para una medición más exacta será preferible, en algunos casos, seleccionar y aislar objetos (para asegurarse de que no hay interferencias) y, en todos los casos, tomar las mediciones sobre un visor ortogonal que muestre una vista sin deformaciones del objeto.

La medición de luz depende de la radiosisidad y para que dé un resultado correcto debe reiniciarse el proceso en caso de que se haya hecho algún cambio. Si, por ejemplo, asignamos a un objeto un material con un valor de brillo de 128 y activamos radiosisidad la medición dará un valor de reflectancia de 50% apro-

ximadamente. Si cambiamos este valor a 64 y volvemos a medir dará el mismo resultado, 50%. Para que dé el valor correcto, 25% aproximadamente, hay que rehacer el cálculo de Radiosisidad.

Otro modo de analizar la iluminación es a través del cuadro de diálogo de *Control de Exposición*. Escoger un tipo de medición (Luminancia o Iluminancia) y fijar los valores mínimos o máximos de modo que se aproximen a las cantidades extremas de la escena. Hacer también que la Escala Física se corresponda con el valor de intensidad de la luz más brillante. La barra de colores reflejará la distribución de valores según esta escala, del Mínimo al Máximo. Al activar el Control de Exposición la escena mostrará estos colores en la escena lo que facilita un análisis visual de la iluminación.

## Análisis de la iluminación con *mental ray*

### Recursos

La versión *3ds Max Design* (no la versión *3ds Max*) incluye la tecnología *Autodesk Exposure* que incorpora una herramienta para hacer mediciones de iluminación cuando se utiliza el motor de *render* de *mental ray* y el método de cálculo FG (*Final Gather*) y luces de tipo *mr Sun* y *mr Sky*. También se pueden utilizar luces fotométricas pero preferiblemente para la iluminación interior. No es recomendable utilizar *mr Sky Portals* pues aunque sus resultados son físicamente correctos su funcionamiento no está validado por los organismos que menciono a continuación y, por otra parte, a menudo se ajustan sus parámetros de un modo que den resultados interesantes pero que no se corresponden exactamente con situaciones reales.

Los resultados pueden ser utilizados, en Estados Unidos, por arquitectos interesados en obtener certificaciones LEED (*Leadership in Energy and Environmental Design*), un sistema desarrollado por el *US Green Building Council* (ver [www.usgbc.org](http://www.usgbc.org)). En la UE no existe por ahora, que yo sepa, un reconocimiento si-



milar para este programa aunque es posible relacionarlo con otras normativas. En cualquier caso es un recurso útil para el diseño aunque sus resultados no estén validados oficialmente en la UE.

Los resultados pueden ser más fiables si se utiliza un modelo de luz celeste adecuado, como el *Perez All Weather* y datos de la escena obtenidos de los archivos *Weather Data* que pueden obtenerse de [www.eere.energy.gov](http://www.eere.energy.gov). Ver el apartado sobre luces diurnas del capítulo 5 sobre luces y sombras.

Autodesk recomienda por añadidura hacer que el *ground color* sea 0,0,0 para evitar reflejos del suelo virtual que pudieran afectar al resultado. Y crear un entorno con edificios simples pero que reflejen luz hacia la escena con valores de reflectancia medios, aunque no sean visibles en la imagen. En cualquier caso, todos los valores de reflectancia de los elementos arquitectónicos deben estar ajustados de un modo realista pues influirán en la iluminación global. Ver mi libro sobre Simulación de Materiales para más información sobre este tema o buscar por Internet valores de reflectancia de materiales corrientes que pueden obtenerse de muchas fuentes.

### Precisión de los análisis

El mejor programa de simulación de la iluminación sigue siendo, muy probablemente, *Radiance*, al que ya me he referido varias veces. En la documentación anexa a los sistemas de iluminación de sus programas, los propios desarrolladores y asesores de Autodesk lo reconocen así y citan un informe de 2006 en el que se analizaron del orden de 200 programas de simulación de diversos tipos, provenientes de 27 países y en los que las referencias principales seguían siendo a este programa.

Las razones principales, que también hemos visto, es que, desde sus inicios, *Radiance* se basó en datos físicos correctos e incluso desarrolló técnicas especiales, que también hemos visto y que dieron lugar al propio formato HDR, además de otras técnicas de simulación, para asegurar que los procedimientos de simulación se adecuaban a los requisitos de procesos físi-

cos correctos. Por añadidura, es un programa flexible que puede incorporar “con facilidad”, otras técnicas que puedan interesar a sus usuarios. Las comillas de la frase anterior aluden a su principal inconveniente para la mayoría de los usuarios arquitectos o diseñadores. *Radiance* no cuenta con un interface amigable y está preparado para correr bajo Unix. Las versiones que intentan adaptarse a Windows lo hacen con enormes dificultades, algo de lo que puedo, personalmente, dar fé.

Por el contrario, los programas más (relativamente) sencillos de utilizar, pueden desviarse considerablemente de las mediciones físicas exactas. Esta observación va dirigida a quienes necesiten llevar a cabo simulaciones precisas. Los métodos que aquí se describen es de suponer que están adecuadamente calibrados, tal como se desprende de la documentación y de las pruebas que he citado. Pero hay un margen de inexactitud que puede ser irrelevante para casos más o menos corriente pero que debe tenerse en cuenta si se trata de llevar a cabo análisis precisos.

La documentación proporcionada por Autodesk incluye algunos análisis de casos concretos, con gráficos que muestran los desplazamientos relativos que se producen en determinados casos relevantes para el análisis energético, como el de ventanas protegidas por persianas venecianas. Ver el documento *3ds\_max\_design-exposure\_validation.pdf* al que se puede acceder directamente desde el *Lighting Analysis Assistant*, presionando el botón “Learn more” que se encuentra en la parte inferior del panel.

Una alternativa interesante es utilizar el programa *Daysim*, basado en *Radiance*. Los resultados proporcionados por *Daysim* son muy similares a los de *Radiance*, principalmente para cielos cubiertos. Para cielos despejados pueden variar más debido a que *Daysim* interpola valores de la contribución solar directa a partir de puntos vecinos.

Es necesario tener en cuenta que tanto la geometría como la definición de materiales deben cumplir ciertos requisitos para evitar determinados errores.



La **geometría** de los elementos que componen la escena debe haberse creado de tal modo que se eviten fugas de luz. Aunque no sea muy habitual, hay modelos que están generados de un modo bastante chapucero lo que puede llevar a que aparezcan “grietas”, zonas abiertas por donde puede pasar la luz exterior, lo que alterará las mediciones interiores.

Además, los elementos que rodeen a un edificio, el plano, edificios adyacentes, etc, también deben modelarse aunque no aparezcan en la imagen y deben recibir asignaciones de materiales adecuadas pues la luz que rebota en ellos contribuirá, de modo importante, como ocurre en la realidad, a la iluminación.

Por lo que respecta al **suelo** del exterior tener en cuenta que si se utiliza un suelo virtual, que se configura desde *mr Sky Parameters*, hay que hacer que su color (*ground color*) sea negro para que los rayos que se reflejan desde el suelo no contribuya a la iluminación general. Y, por otro lado, dado que los planos horizontales que rodean al edificio sí contribuirían a la iluminación, convendrá crear un plano ad hoc con valores de reflectancia adecuados al caso. Si el plano no estaba previsto en el modelo, crearlo de todos modos, con una anchura suficiente como para que los rayos rasantes que reboten en él alcancen el edificio. Valores de reflectancia característicos son los correspondientes a una reflectancia de un 20% es decir, RGB 0.20, 0.20, 0.20.

Los **materiales** deben estar definidos con valores de reflectancia que correspondan a materiales reales. La mejor recomendación es utilizar tablas con valores de reflectancia provenientes de fuentes fiables y asegurarse de que las asignaciones se mueven en un rango cercano a estos valores. En cualquier caso no utilizar valores extremos, pues no son reales. Una pared “blanca” no refleja el 100% de la luz que recibe sino, como máximo, un 90%. Y una pared “negra” tampoco refleja un 0% sino, como mínimo, un 5%. Esto quiere decir que sus especificaciones, en coma flotante, serían RGB 0.90, 0.90, 0.90 y 0.05, 0.05, 0.05 respectivamente.

## Parámetros principales

Cuando se activa el sistema de análisis de la iluminación, como haremos en el ejemplo que sigue, se abre el panel **Lighting Analysis Assistant**. Es un panel sencillo con cuatro secciones que resumo brevemente a continuación. Para una descripción más extensa consultar la ayuda del programa:

- a) **General**. Muestra la configuración activa (FG, etc) y un botón, *Edit*, que permite editarla sin necesidad de salir del asistente. Si hay valores incorrectos se resaltarán en negrita y se indicará que son inválidos. En la gran mayoría de los casos no habrá que hacer nada puesto que los valores mostrados corresponderán a configuraciones corrientes. En la parte inferior hay una codificación de color que, por defecto, es lineal y tiene un rango de 0.0 a 10.000 luxes. Convendrá cambiarla haciendo que el valor máximo coincida con la iluminación máxima de la escena. Pero esto no afecta al cálculo, solo a los colores que se utilizan para representar los niveles de luz, por lo que no tiene mayor importancia.
- b) **Lighting**. Muestra los valores de las luces presentes en la escena y, en el caso de un sistema de luz diurna, las coordenadas geográficas, la fecha y la hora. Como en el caso anterior, incluye botones de acceso directo a estas configuraciones sin necesidad de salir del asistente.
- c) **Materials**. Indica si hay materiales inválidos en la escena e incluye también botones de acceso directo para editarlos si fuera el caso.
- d) **Analysis Output**. Tiene dos secciones. En la primera, *Light Meter*, se incluye un botón para crear un *Light Meter* (puede crearse independientemente desde el menú *Lighting Analysis/ Create/ Light Meter*). El *Light Meter* se crea como un *sky portal*, pinchando y arrastrando sobre la escena. El asistente incluye también un botón, *Calculate All Light Meters Now*, para activar el cálculo (y otro para exportar los datos) pero



lo haremos desde fuera. La segunda sección, *Image Overlay*, permite crear un efecto de postproducción, al presionar el botón *Create Image Overlay Render Effect*, para superponer a la imagen generada al hacer un *render* los valores medidos.

Una vez creado este efecto, editarlo y comprobar que, entre otras cosas, los valores de salida se pueden mostrar en luxes (iluminancias) o en candelas/m<sup>2</sup> (luminancias). Tener también en cuenta que lo que se muestra en este caso depende de la vista pues es el resultado de una intersección entre rayos enviados desde la cámara y las superficies que estos rayos encuentran en su recorrido.

- e) Si se modifican cualquiera de los valores anteriores hay que presionar el botón *Update*, en la parte inferior del panel, para que se actualicen.

Cuando se crea un *Light Meter Helper* se crea una retícula en la escena que se puede editar desde el panel *Modify*. Este panel tiene dos secciones. En *Parameters* se pueden cambiar las dimensiones y el número de segmentos (lo que implica un mayor o menor número de nodos). Y en *Display* hay opciones para desactivar el que se muestren las flechas de orientación del plano (*Show Orientation*), el color que indica las diferencias de iluminación (*Show Pseudo Color*), el texto con los valores medidos (*Show Viewport Text*). Y, lo más importante, los valores que se mostrarán: *Total Illuminance*, *Direct Illuminance*, *Indirect Illuminance*, *Daylight Factor*. Los valores de los tres primeros se dan en luxes y los del cuarto en porcentaje.

Para activar el cálculo hay que pulsar, en este mismo panel, el botón *Calculate All Light Meters*. Y, si se pulsa el botón *Export To CSV File* se pueden enviar los datos de salida a Excel.

### Ejemplo 1. Galería. Iluminación diurna

Partimos de una escena como la de la figura adjunta que representa un espacio exterior

con una pequeña terraza y un espacio interior con un gran ventanal. El análisis se referirá al espacio exterior con luz diurna, al espacio interior con luz diurna y al espacio interior con luces artificiales.

1. Crear un sistema de luz diurna. Configurarlo para una localización geográfica, una fecha y una hora determinadas. En este ejemplo está referido a Barcelona, el 1 de septiembre a las 10:00.
2. Cambiar el tipo de sol a *mr Sun* y el tipo de luz celeste a *mr Sky*.
3. Activar el cálculo de *Final Gather* con calidad media y un par de rebotes. El control de exposición que he utilizado en este ejemplo es *mr photographic control exposure* con EV 15.0. Hacer un *render* de comprobación que deberá dar un resultado similar al de la figura adjunta.
5. Crear un *Light Meter Helper*. Esto puede hacerse de dos modos: abriendo previamente, desde el menú *Lighting Analysis*,

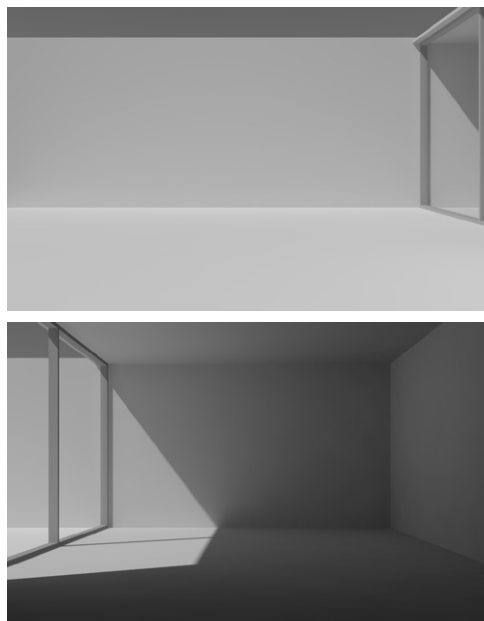


Figura 6.50 Análisis de la iluminación. Ejemplo 1. Vista exterior (a) e interior (b) del ejemplo ("galería") descrito en el texto

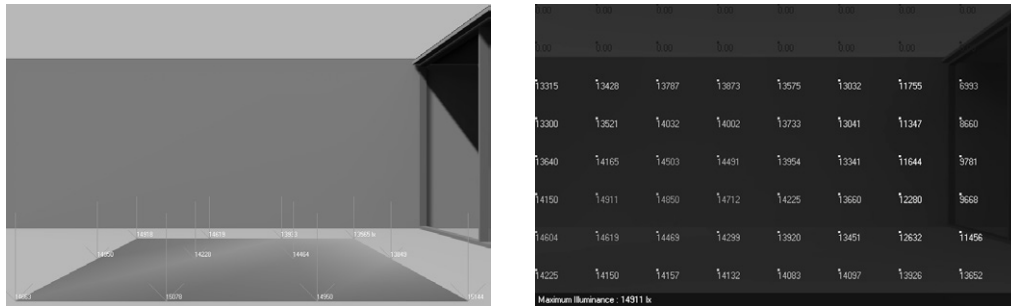


Figure 6.51 shows the lighting analysis for an interior space. Part (a) is a 3D render of a room with a light meter overlay showing illuminance values at various nodes. Part (b) is a 3D render of the same room with illuminance values superimposed on the surfaces. Part (c) is a table of the values obtained.

Name	ID	Loc	Fecha	Dir	InDir	Total	Und	DF	Pos XYZ	Und	Orient
LightM1	1	Bcn	1 sep, 10:00	0	14.663	14.663	lx	0.0	-722.75, -106.11, 8	mm	0,0,1
LightM1	2	Bcn	1 sep, 10:00	0	15.078	15.078	lx	0.0	-619.74, -106.11, 8	mm	0,0,1
LightM1	3	Bcn	1 sep, 10:00	0	14.950	14.950	lx	0.0	-516.74, -106.11, 8	mm	0,0,1
LightM1	4	Bcn	1 sep, 10:00	0	15.144	15.144	lx	0.0	-413.73, -106.11, 8	mm	0,0,1
LightM1	5	Bcn	1 sep, 10:00	0	14.950	14.950	lx	0.0	-722.75, 134.52, 8	mm	0,0,1
LightM1	6	Bcn	1 sep, 10:00	0	14.228	14.228	lx	0.0	-619.74, 134.52, 8	mm	0,0,1
LightM1	7	Bcn	1 sep, 10:00	0	14.464	14.464	lx	0.0	-516.74, 134.52, 8	mm	0,0,1
LightM1	8	Bcn	1 sep, 10:00	0	13.849	13.849	lx	0.0	-413.73, 134.52, 8	mm	0,0,1
LightM1	9	Bcn	1 sep, 10:00	0	14.918	14.918	lx	0.0	-722.75, 375.15, 8	mm	0,0,1
LightM1	10	Bcn	1 sep, 10:00	0	14.619	14.619	lx	0.0	-619.74, 375.15, 8	mm	0,0,1
LightM1	11	Bcn	1 sep, 10:00	0	13.933	13.933	lx	0.0	-516.74, 375.15, 8	mm	0,0,1
LightM1	12	Bcn	1 sep, 10:00	0	13.565	13.565	lx	0.0	-413.73, 375.15, 8	mm	0,0,1

Figura 6.51 Análisis de la iluminación. Ejemplo 1. Vista interior: a) Vista del Light Meter con valores de iluminancia en los nodos, b) Vista de la salida de render con superposición de valores de iluminancia en puntos de las superficies, c) Tabla de los valores obtenidos

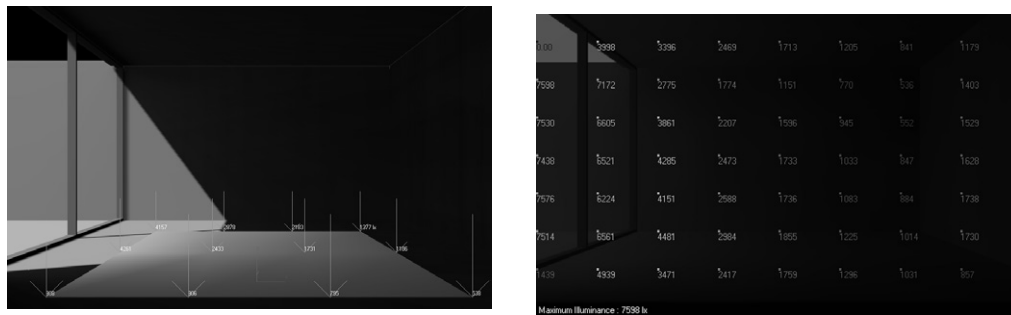


Figure 6.52 shows the lighting analysis for an interior space. Part (a) is a 3D render of a room with a light meter overlay showing illuminance values at various nodes. Part (b) is a 3D render of the same room with illuminance values superimposed on the surfaces. Part (c) is a table of the values obtained.

Name	ID	Loc	Fecha	Dir	InDir	Total	Und	DF	Pos XYZ	Und	Orient
LightM1	1	Bcn	1 sep, 10:00	0	908	908	lx	6.06	-152.69, -106.11, 8	mm	0,0,1
LightM1	2	Bcn	1 sep, 10:00	0	906	906	lx	6.04	-49.69, -106.11, 8	mm	0,0,1
LightM1	3	Bcn	1 sep, 10:00	0	795	795	lx	5.30	53.32, -106.11, 8	mm	0,0,1
LightM1	4	Bcn	1 sep, 10:00	0	538	538	lx	3.58	156.32, -106.11, 8	mm	0,0,1
LightM1	5	Bcn	1 sep, 10:00	0	4.261	4.261	lx	28.41	-152.69, 134.52, 8	mm	0,0,1
LightM1	6	Bcn	1 sep, 10:00	0	2.433	2.433	lx	16.22	-49.69, 134.52, 8	mm	0,0,1
LightM1	7	Bcn	1 sep, 10:00	0	1.731	1.731	lx	11.54	53.32, 134.52, 8	mm	0,0,1
LightM1	8	Bcn	1 sep, 10:00	0	1.186	1.186	lx	7.91	156.32, 134.52, 8	mm	0,0,1
LightM1	9	Bcn	1 sep, 10:00	0	4.157	4.157	lx	27.71	-152.69, 375.15, 8	mm	0,0,1
LightM1	10	Bcn	1 sep, 10:00	0	2.878	2.878	lx	19.19	-49.69, 375.15, 8	mm	0,0,1
LightM1	11	Bcn	1 sep, 10:00	0	2.183	2.183	lx	14.56	53.32, 375.15, 8	mm	0,0,1
LightM1	12	Bcn	1 sep, 10:00	0	1.377	1.377	lx	9.18	156.32, 375.15, 8	mm	0,0,1

Figura 6.52 Análisis de la iluminación. Interior: a) Vista del Light Meter con valores de iluminancia en los nodos, b) Vista de la salida de render con superposición de valores de iluminancia en puntos de las superficies, c) Tabla de los valores obtenidos





el *Lighting Analysis Assistant* y, desde la sección *Analysis Output*, pulsando el botón *Create a Light Meter* o bien, desde el mismo menú, entrando en *Create / Light Meter*. En cualquier caso, todo lo que hay que hacer es pinchar y arrastrar sobre el plano de planta con lo que se creará una retícula con flechas apuntando hacia abajo en cada nodo. Luego comprobar que está situado ligeramente por encima de la superficie seleccionada para que no haya interferencias en la medición. Si en el resultado final apareciera algún valor 0.0 lo más probable es que sea debido a algún obstáculo.

6. Editar el *Light Meter* desde el panel *Modify* y configurarlo como sigue:
  - a) En *Parameters*, cambiar su tamaño y su posición y su número de segmentos. El cambio en el número de segmentos afectará al número de lecturas, pues se incluirá una lectura por cada vértice.
  - b) En *Display*, en el grupo *Values to Display* escoger los valores a mostrar de las cuatro opciones disponibles (*Total Illuminance*, *Direct Illuminance*, *Indirect Illuminance*, *Daylight Factor*).
7. Presionar el botón *Edit* para volver al asistente. En la sección *General*, comprobar los valores de cálculo de iluminación por si hubiera que modificarlos (los valores incorrectos aparecerán resaltados en negrita) y, si se quiere utilizar el mapa de pseudo color, cambiar el rango en función de los primeros valores obtenidos y, si esto no interesa, no hacer nada. En la sección *Lighting* comprobar los valores que estamos utilizando por si hubiera que modificarlos. En la sección *Materials*, comprobar que no hay avisos sobre materiales inválidos y, si los hubiese, substituir el material por un material adecuado (*Arch&Design* será la mejor opción, como ya hemos visto más arriba). Y, en *Analysis Output*, si se prefiere, activar la opción *Create Image Overlay Render Effect* para que los valores obtenidos se escriban sobre la imagen de salida.

8. Presionar el botón *Calculate All Light Meters*. Tras unos instantes, se efectuará el cálculo y los resultados se mostrarán sobre los nodos del *Light Meter*, tal como se muestra en la figura adjunta. En el grupo *Display* se pueden cambiar las cuatro opciones citadas que se modificarán inmediatamente.

Si todo lo que nos interesa es comprobar la iluminación que se recibe en determinados puntos, esto es todo. Si queremos llevar a cabo un análisis más preciso o necesitamos procesar los datos de otro modo, pulsar el botón *Export to CSV File*. Seleccionar una carpeta de destino y dar un nombre al archivo. Luego abrir Excel, y abrir el archivo que mostrará todos sus datos en una sola columna. Activar el comando de menú *Datos / Desde Texto*. Seleccionar el archivo que hemos generado. Se abrirá un panel especial "Asistente para importar texto". En la primera sección, en Tipo de datos, escoger "Delimitados". en la segunda, en Separadores, escoger "Coma". Y en la tercera, en Formato de los datos en columnas, escoger "General". Los datos aparecerán ordenados como en la figura adjunta que, como se puede ver, muestran los valores de las cuatro opciones *Total Illuminance*, *Direct Illuminance*, *Indirect Illuminance* y *Daylight Factor*.

Para que se muestre el *Daylight Factor* hay que editar el sistema de luz diurna y, en *mr Sky Parameters / Sky Model*, seleccionar CIE. Y en la sección que así se abrirá, *mr Sky: CIE Parameters*, cambiar el valor de *Diffuse Horizon* según los criterios que se dan en el siguiente párrafo y hacer que el valor de *Direct Normal Illuminance* sea 0.0 y la opción seleccionada en la parte inferior sea *Overcast Sky*. Si luego se activa mostrar la opción *Day Light Factor*, estos valores se mostrarán en la pantalla y también se incluirán en la tabla de exportación si se utiliza. Si esto no se hace así no aparecerán en pantalla y el valor exportado será 0.0.

El valor asignado al *Diffuse Horizon* depende de la localización. Para averiguar este dato hay que utilizar tablas adecuadas o recursos que se encuentran en Internet como



el “Design Sky”. Este recurso representa la iluminancia horizontal que se da en más de un 85% durante el periodo comprendido entre las 9:00 y las 17:00, es decir que supone un valor medio de referencia suficiente para cálculos simplificados. Y este valor depende de la latitud. Los valores correspondientes a una determinada latitud se pueden obtener a partir de una serie de fórmulas desarrolladas por Peter Tregenza. Los valores predeterminados de 10.000 luxes corresponden aproximadamente a una zona relativamente al norte, más o menos a la altura del Reino Unido. Para el Ecuador deberían ser de 18.000 luxes. Y para países como España, en torno a 15.000.

Si se asigna al horizonte un valor adecuado se pueden interpretar los resultados como correspondientes a los niveles internos de iluminación del modelo analizado y, por tanto, se puede estimar el *Daylight Factor* pero con la salvedad de que estos valores tendrían sentido únicamente para una situación en que el cielo estuviera completamente cubierto o hubiera obstáculos que impidieran la entrada directa del sol.

Si después del cálculo los *Light Meters* aparecen completamente blancos (lo que dificultará leer los valores) lo más probable es que esto se deba a que los rangos de pseudocolor son inadecuados. Para corregir esto, editar el *Assistant* y modificar el rango.

Si se utilizan materiales con texturas será necesario calibrar adecuadamente las superficies y esto es complicado. Por esta razón, lo más recomendable es asignar a toda la es-

cena un único material de reflectancia media utilizando el recurso *material override* que ya hemos visto y que se encuentra en *Render Setup/Processing/ material Override*.

Las figuras adjuntas muestran resultados obtenidos para un mismo modelo y con una misma iluminación pero tomados en el exterior y en el interior, respectivamente.

Según los casos convendrá escoger fechas y horas adecuadas para el análisis. Por ejemplo, puede interesar analizar la iluminación a las 9:00, 12:00 y 15:00 y en los solsticios (21 de junio y 21 de diciembre) y equinoccios (21 de marzo y 22 de septiembre).

## Ejemplo 2. Galería. Iluminación con luces artificiales

El mismo tipo de cálculo puede repetirse para luces fotométricas. En la última imagen de este grupo se muestran los resultados obtenidos para el mismo espacio pero desactivando la luz diurna y añadiendo luces fotométricas.

Las luces utilizadas tienen una intensidad de 100 cd. El tipo de distribución está basado en un archivo IES genérico semidistribuido, que corresponde a uno de los tres tipos que he mostrado en el capítulo anterior y que están generado con los métodos allí descritos pero que pueden copiarse y aplicarse con facilidad a cualquier escena.

El control de exposición en este caso es de 4.0. El cálculo se ha hecho, como en los ejemplos anteriores con FG de calidad media y un par de rebotes.

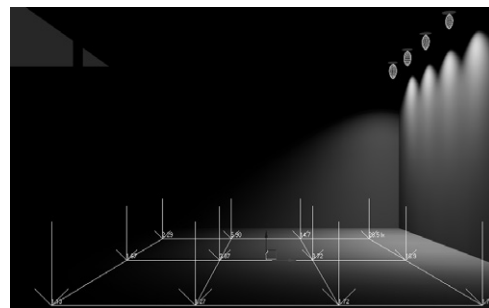


Figura 6.53 Análisis de la iluminación. Interior iluminado con luces fotométricas: a) Vista del *Light Meter* con valores de iluminancia en los nodos, b) Vista de salida con superposición de valores de iluminancia





## Referencias

- Aila T., Laine, S. 2009. "Understanding the efficiency of ray traversal on GPUs". En *Proceedings of High Performance Graphics* (2009), pp 145.
- Akerlund, O.; Unger, M.; Wang, R. 2007. "Precomputed visibility cuts for interactive relighting with dynamic BRDFs". En *Proceedings of Pacific Graphics* (2007), pp 161–170.
- Akimoto, T. et al. 1991. "Pixel selected Ray Tracing". *IEEE CG&A*, jul 1991, p 15.
- Appel, A. 1968. "Some techniques for machine rendering of solids". *AFIPS Conference Proc*, 32, pp 37-45.
- Arikan, O.; Forsyth, D.A.; O'Brien, J.F. 2005. "Fast and Detailed Approximate Global Illumination by Irradiance Decomposition". *Siggraph 2005*.
- Arvo, J. 1986. "Backward Ray Tracing". *Developments in Ray Tracing. SIGGRAPH Course Notes*, vol 12.
- Arvo, J., and D. Kirk. 1990. "Particle transport and image synthesis". *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4), pp 63–66.
- Arvo, J. 1993. "Transfer equations in global illumination". *Global Illumination, SIGGRAPH '93 Course Notes*, vol 42.
- Arvo, J. 1995. *Analytic Methods for Simulated Light Transport*. Ph.D. thesis, Yale University.
- Ashikhmin, M. 2002. "A tone mapping algorithm for high contrast images". En *Proc. of Eurographics Workshop on Rendering 2002*, pp 145-156.
- Autodesk: 3ds Max Design White Papers. Ver <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=11360846>.
- Barten, P. 1992. "Physical model for the contrast sensitivity of the human eye". En *Proc. SPIE*, vol. 1666, pp 57-72.
- Barten, P. 1993. "Spatio-temporal model for the contrast sensitivity of the human eye and its temporal aspects". En *Proc. SPIE*, vol 1913-01.
- Barten, P. 1999. *Contrast Sensitivity of the human eye and its effects on image quality*. HV Press, Eindhoven.
- Bikkery, J.; Reijersez, R. 2009. "A Precalculated Point Set for Caching Shading Information". *Eurographics 2009*.
- Bikker, J; Van Schijndel, J. 2013. "The Brigade Renderer: A Path Tracer for Real-Time Games". *International Journal of Computer Games Technology*, vol 2013.
- Bloch, C. *The HDRI Handbook. High Dynamic Range Imaging for Photographers and CG Artist*. O'Reilly, 2008.
- Bredow, R. 2002. "RenderMan on Film", *Course 16: RenderMan in Production*. ACM SIGGRAPH 2002 Course Notes.
- Brunger, A. P.; Hooper, F. C. "Anisotropic sky radiance model based on narrow field of view measurements of shortwave radiance". *Solar Energy* (1993).
- Bunnell, M. 2005. "Dynamic ambient occlusion and indirect lighting". *GPU Gems 2* 2 (2005), pp 223-233, 2.
- Carpenter, L., Fournier, A. & Fusell, D. 1982. "Computer rendering of stochastic models". *Comm. ACM*, 25 (6), pp 371-84.
- Christensen, P. H. 2003. "Adjoints and importance in rendering: an overview". *IEEE Transactions on Visualization and Computer Graphics*, 9(3), pp 329–40.
- Christensen, P. 2008. "Point-based approximate color bleeding". *Tech. rep., Pixar Tech. Memo #08-01*. 2008. 1,2,9.
- CIBSE: "Surface Reflectance and Colour". Ver <http://www.cibse.org/index.cfm?go=publications.view&item=164>.
- CIE (Commission Internationale de l'Eclairage). Ver el sitio [www.cie.co.at](http://www.cie.co.at).
- CIE-110-1994. "Spatial distribution of daylight - luminance distributions of various reference skies". *Tech. rep., International Commission on Illumination*, 1994.
- Cohen, M.F. & Greenberg, D.P. 1985. "The Hemisphere. A Radiosity Solution for Complex Environments". *SIGGRAPH*, 85, 31.
- Cohen, M.F., Greenberg, D.F. & Immel, D. S. 1986.



- "An efficient radiosity approach for realistic image synthesis". *IEEE Computer Graphics and Applications*, vol 6, n 2, pp 26-35.
- Collins, S. 1994. "Adaptive splatting for specular to diffuse light transport". *Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, pp 119-35.
- Cook, R.L. & Torrance, K.E. 1982. "A Reflectance Model for Computer Graphics". *Computer Graphics*, 15: 3, pp 307-316.
- Cook, R.L. 1984. "Shade Trees". *Computer Graphics*, vol 18, no 3, pp 223-231.
- Cook, R.L., Porter, T., Carpenter, L. 1984. "Distributed Ray Tracing". *SIGGRAPH 84*.
- Cook, R.L. 1986. "Stochastic sampling in computer graphics". *ACM Trans. on Computer Graphics*, 5 (1), pp 51-72.
- Cornsweet, T. 1970. *Visual Perception*. Harcourt Brace, San Diego.
- Crow, F.C. 1977. "Shadow Algorithms for Computer Graphics". *Siggraph 77*, pp 242-247.
- CVRL (Color & Vision Research Laboratory and Database). <http://www.cvrl.org/>. Página web de este laboratorio del University College London. Base de datos disponible desde 1995.
- Dachsbacher, C.; Stamminger, M. 2005. "Reflective Shadow Maps". En: *Proceedings SI3D '05*, pp 203-231.
- Debevec, P.; Malik, J. 1997. "Recovering High Dynamic Range Radiance Maps from Photographs". *Proceedings Siggraph97*, pp 369-378.
- Debevec, P. 1998. "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography". *SIGGRAPH98*. Disponible en <http://www.debevec.org/Research/IBL/>.
- Dobashi, Y., Yamamoto, T., Nishita, T. 2002. "Interactive rendering of atmospheric scattering effects using graphics hardware". *Proceedings of the conference on Graphics hardware*, 2002.
- Drago, F.; Myszkowski, K.; Annen, T.; Chiba, N. 2003. "Adaptive Logarithmic Mapping For Displaying High Contrast Scenes" (2003). *Computer Graphics Forum*, vol 22, pp 419-426.
- Durand, F.; Dorsey, J. 2002. "Fast bilateral filtering for the display of high-dynamic-range images". *ACM Trans. Graph* (special issue SIGGRAPH 2002) 21, 3, pp 257-266.
- Dutr , P.; Willems, Y.D. 1994. "Importance-Driven Monte Carlo Light Tracing". *Fifth Eurographics Workshop on Rendering*, pp 185-194, Darmstadt.
- Dutr , P.; Bala, K.; Bekaert, P. 2006. *Advanced Global Illumination*. A.K.Peters, Massachusetts.
- Ecotec. Ver Autodesk: Ecotect Wiki, <http://squ1.org/wiki/>.
- Energy Plus: Weather Data Files. Ver [http://www.eere.energy.gov/buildings/energyplus/cfm/weather\\_data.cfm](http://www.eere.energy.gov/buildings/energyplus/cfm/weather_data.cfm).
- Fajardo, M. 2001. "Monte Carlo ray tracer in action". En: *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*. Siggraph 2001 Course 29, pp 151-162.
- Fattal, R., Lischinski, D., Werman. 2002. "Gradient domain high dynamic range compression". *ACM Trans. Graph.* (special issue SIGGRAPH 2002) 21, 3, pp 249-256.
- Foley, J. D., A. Van Dam, S. K. Feiner, and J. F. Hughes. 1990. *Computer Graphics: Principles and Practice*. Reading, Massachusetts: Addison-Wesley. 3<sup>a</sup> ed: 2013.
- Fournier, A., Fussell, D., Carpenter, R.L. 1982. "Computer rendering of stochastic models". *CACM*, vol.25, pp 371-384.
- Freeman, M. 2008. *Mastering High Dynamic Range Photography*. Trad. Esp.: *Manual de fotograf a de gama din mica amplia (HDR)*. Barcelona, Blumen, 2009.
- Frisby, J.P. Seeing. 1979. *Illusion, Brain and Mind*. Trad.Esp: *Del ojo a la visi n*. Madrid, Alianza, 1987.
- Glassner, A.S. 1984. "Space subdivision for fast ray tracing". *IEEE Computer Graphics and Applications*, 14 (10), pp 15-22.
- Glassner, A.S. 1995. *Principles of Digital Image Synthesis*. Kaufmann, San Francisco.
- Gonzalez, R.C. & Woods, R.E. 1992. *Digital Image Processing*. Reading, Mass., Addison-Wesley.
- Goral, C. M., Torrance, K. E., Greenberg, D. P. and Battaile, B. 1984. "Modeling the Interaction of Light Between Diffuse Surfaces", *Computer Graphics*, vol 18, no 3, pp 213-222.
- Green R. 2003. "Spherical Harmonic Lighting: The Gritty Details". En <http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf>.
- Greenberg, D.P. 1989. "Light Reflection Models for

- Computer Graphics". *Science*, vol 244, n 14, pp166-173.
- Greenberg, D. P., K. E. Torrance, P. S. Shirley, J. R. Arvo, J. A. Ferwerda, S. Pattanaik, E. P. F. La-fortune, B. Walter, S.-C. Foo, and B. Trumbore. 1997. "A framework for realistic image synthe-sis". *Proceedings of SIGGRAPH '97, Computer Graphics Proceedings, Annual Conference Se-ries*, pp 477-94.
- Hachisuka, T. 2005. "High Quality Global Illumi-nation Rendering Using Rasterization". *GPU-Gems 2*, pp 615-634.
- Hall, E.L. 1979. *Computer Image Processing and Recognition*. New York, Academic Press.
- Hanrahan, P., and J. Lawson. 1990. "A language for shading and lighting calculations". En: F. Baskett (Ed.), *Computer Graphics (SIGGRAPH '90 Pro-ceedings)*, vol 24, pp 289-98.
- Hanrahan, P., Salzman, D, Aupperle, L. 1991. "A Rapid Hierarchical Radiosity Algorithm". *Com-puter Graphics (Proc. Siggraph)*, vol 25, n 4, july 1991, pp 197-206.
- Hanrahan, P.; Ramamoorthi, R. 2001. "An Efficient Representation for Irradiance Environment Maps". *Siggraph '01. Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, pp 497-500.
- Hasan, M.; Pellacini, F.; Bala, K. 2007. "Matrix row-column sampling for the many-light problem". *ACM Transactions on Graphics*. 26, 3, 26, pp 1-10.
- Hasan, M.; Pellacini, F.; Bala, K. 2007. "Matrix row-column sampling for the many-light problem". *ACM Transactions on Graphics* 26, 3 (2007), 26. (Proc. Siggraph).
- Hecht, S. 1924. "The Visual Discrimination of Inten-sity and the Weber-Fechner Law," *J. Gen Physi-ol.*, vol 7, p 241.
- Hecht, S.; Shlaer, S.; Pirenne, H. 1942. "Energy, Quanta and Vision". *The Journal of General Physiology*, vol 25, pp 819-840.
- Heckbert, P.S. & Hanrahan, P. 1984. "Beam tracing Polygonal Objects". *Computer Graphics*, 18, (3), pp 119-145.
- Heckbert, P. S. 1990b. "Adaptive radiosity tex-tures for bidirectional ray tracing". En *Computer Graphics (Proceedings of SIGGRAPH '90)*, vol 24, pp 145-54.
- Heidmann, T. 1991. "Real Shadows in Real Time," *IRIS Universe*, (18) pp 28-31.
- Hoberock, J.; Jia, Y. 2013. "High Quality Ambient Occlusion". *GPU Gems 3*, 12.
- Hoffman, N., Preetham, A.J. 2002. "Rendering out-door light scattering in real time". *Game Devel-opers Conference*, 2002.
- Hopkinson, R.G.; Petherbridge, P.; Longmore, J. 1966. *Daylighting*. London, Heinemann
- Illuminating Engineering Society of North America Computer Committee. 1986. *IES recommended standard file format for electronic transfer of pho-tometric data*. IES LM- 63-1986.
- Illuminating Engineering Society of North America. 2002. *IESNA standard file format for electronic transfer of photometric data*. BSR/IESNA Publi-cation LM-63-2002. Ver www.iesna.org.
- Immel, D.S., Cohen, M.F., Greenberg, D.P. 1986. "A Radiosity Method for Non-Diffuse Environ-ments". *Computer Graphics (Proc. Siggraph)*, vol 20, n 4, aug. 1986, pp 133-142.
- Jacobs, A. 2007. *High Dynamic Range Imaging and its Application in Building Research; Advances in Building Energy Research*, James & James, London.
- Jensen,H. 1995. "Importance Driven Path Tracing Using the Photon Map," *Rendering Techniques '95 (Proc. Sixth Eurographics WorkshopRender-ing)*, pp 326-335.
- Jensen, H.W.; Christensen, N.J. 1995. "Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects". *Computers & Graphics*, vol. 19 (2), pp 215-224, march.
- Jensen, H. W. 1996. "Global illumination using pho-ton maps". *Rendering Techniques '96: 7th Eu-rographics Workshop on Rendering*, pp 21-30.
- Jensen, H. W. 1996b. "Rendering caustics on non-Lambertian surfaces". *Graphics Interface '96*, pp 116-21.
- Jensen, H.W. 2001. *Realistic Image Synthesis Us-ing Photon Mapping*. A.K.Peters.
- Jensen, H.W (org) et al. 2003. "Monte Carlo Ray Tracing". *Siggraph 2003 Course 44* (parti-cipantes: James Arvo, Phil Dutre, Alexander Kel-ler, Henrik Wann Jensen, Art Owen, Matt Pharr, Peter Shirley).
- Kajiya, J.T. & Kay, T.L. 1986. "Ray tracing complex scenes". *Computer Graphics*, 20 (4), pp 269-78.





- Kajiya, J.T. 1981. "Ray tracing parametric patches". *Computer Graphics*, 16, (3), pp 245-54.
- Kajiya, J.T. 1986. "The rendering equation". *Computer Graphics*, 20 (4), pp 143-50.
- Kay, T., and J. Kajiya. 1986. "Ray tracing complex scenes". *Computer Graphics (SIGGRAPH '86 Proceedings)*, vol 20, pp 269-78.
- Keller A. 1997. "Instant radiosity". En *Proceedings of Siggraph* 31, 3, pp 49-56.
- Kittler, R.; Darula, S. 2002. "CIE general sky standard defining luminance distributions". *Bi-annual Conference of IBPSA*, Ver también *Moon*, pp 11-13.
- Kittler, R.; Kocifaj, M.; Darula, S. 2012. *Daylight Science and Daylighting Technology*. Springer.
- Kollig, T., and A. Keller. 2000. "Efficient bidirectional path tracing by randomized Quasi- Monte Carlo integration". *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pp 290-305. Berlin, Springer-Verlag.
- Lafortune, E., and Y. Willems. 1993. "Bi-Directional Path Tracing". *Proceedings of Computer Graphics '93*, Alvor, Portugal, 1993.
- Lafortune, E. P., and Y. D. Willems. 1996. "Rendering participating media with bidirectional path tracing". *Eurographics Rendering Workshop 1996*, pp 91-100.
- Lafortune, E., and Y. Willems. 1994. "A theoretical framework for physically based rendering". *Computer Graphics Forum* 13(2), pp 97-107.
- Lafortune, E. P., and Y. D. Willems. 1996. "Rendering participating media with bidirectional path tracing". En *Eurographics Rendering Workshop*, 1996, pp 91-100.
- Landis, H. 2002. "Production-Ready Global Illumination". En: *Course 16 notes, SIGGRAPH 2002*.
- LeGrand, Y. 1946, 1956. *Optique Physiologique*. Éditions de la Rev d'Optique, Paris.
- Littlefair, P.J. 1994. "A comparison of sky luminance models with measure data from Garston, United Kingdom". *Solar Energy*, 53, 4, pp 315-322.
- Liu X.; Sloan, P. P.; Shum, H. Y.; Snyder, J. 2004. "All frequency precomputed radiance transfer for glossy objects". En *Proceedings of EGSR* (2004), vol. 3, pp. 337-344.
- Ma V. C. H.; McCool M. D. 2002. "Low latency photon mapping using block hashing". En *Proceedings of Graphics Hardware* (2002), pp 89-99.
- Maletz, D.; Wang, R. 2011. "Importance Point Projection for GPU-based Final Gathering". *Computer GraphicsForum* 30, 4, pp 1327-1336. (Proc. EGSR).
- Maletz, D.; Bowers, J.; Wang, R. 2009. "Reflection Filtering for Interactive Global Illumination in Semi-Glossy Scenes". *Technical Report UM-CS*.
- Mann, S. 1993. "Compositing Multiple Pictures of the Same Scene", *IS&T's 46th Annual Conference*, Cambridge, Massachusetts, May 9-14.
- Marion A. 1987. *Introduction aux Techniques de Traitement d'Images*, Paris. Versión inglesa: *An Introduction to Image Processing*, Chapman and Hall, Tokio, 1991.
- McCann, 2010. "The Ansel Adams Zone System: HDR Capture and Range Compression by Chemical Processing". *Proceedings of SPIE, the International Society for Optical Engineers*.
- McGuire, M.; Luebke, D. 2009. "Hardware-accelerated global illumination by image space photon mapping". En: *Proceedings of HPG '09*, pp 77-89, 2.
- Minnaert, M. *The nature of Light and Color in the Open Air*. Dover, 1954.
- Monte Carlo & Quasi-Monte Carlo Methods. Ver [www.mcqmc.org](http://www.mcqmc.org), portal con enlaces a varias páginas web con información diversa sobre estos métodos.
- Moon, P., Spencer, D. 1942. "Illumination from a non-uniform sky". *Illum Eng.*, 37, 10, pp 707-726.
- Moon, P.; Spencer, D. 1945. "The visual effect of nonuniform surrounds". *Journal of the Optical Society of America* 35, 3, pp 233-248.
- Moon, P. H. 1961. *The Scientific Basis of Illuminating Engineering*, Dover, New York.
- National Research Council of Canada: "Introduction to dynamic daylight metrics". Ver <http://irc.nrc-cnrc.gc.ca/pubs/fulltext/nrcc48669/>.
- Ng, R.; Ramamoorthi, R.; Hanrahan, P. 2003. "All frequency shadows using non-linear wavelet lighting approximation". *ACM Transactions on Graphics* 22, 3, pp 376-381. (Proc. SIGGRAPH).
- Niederreiter, H.; Jau-Shyong Shiue, P. (ed). 1995. *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. New York, Springer-Verlag.

- Nijasure, M.; Pattanaik, S.; Goel, V. 2004. "Real-time global illumination on the GPU. *Journal of Graphics Tools*.
- Parker, S.G.; Bigler, J.; Dietrich, A., et al. 2010. "Optix: A general purpose ray tracing engine. *ACM Transactions on Graphics* 29, 66:1–66:13.
- Pattanaik, S.N.; Mudur, S.P. 1993. "The Potential Equation and Importance in Illumination Computations". *Computer Graphics Forum*, 12:2, pp 131-136.
- Pattanaik, S. N.; Ferwerda, J. A.; Fairchild, M. D.; Greenberg, D. 1998. "A multiscale model of adaptation and spatial vision for realistic image display". En: *Proc. of ACM SIGGRAPH '98*, pp 287-298.
- Perez, R.; Seals, R.; Michalsky, J. 1993. "An all-weather model for sky luminance distribution". *Solar Energy*, 50, 3, pp 235-245.
- Pharr, M.; Humphreys, G. 2004. *Physically Based Rendering*. Morgan Kaufman/Elsevier, Amsterdam.
- Picott, K.P. 1992. "Extensions of the Linear and Area Lighting Models". *IEEE Computer Graphics and App*, march, pp 31-38.
- Polyak, S.L. 1941. *The Retina*. University of Chicago Press, Chicago.
- Poynton, Ch. Ver su página personal, [www.poynton.com](http://www.poynton.com) que incluye varias referencias de todo tipo sobre video digital y *gamma*.
- Pratt, W.K. 1978. Reed 4ª: 2007. *Digital Image Processing*. J.Wiley, New York.
- Preetham, A.J., Shirley, P., Smits, B.E. 1999. "A practical analytical model for daylight". *Proceedings of Siggraph*, pp 91-100, 1999.
- Preetham, A. J. 2003. "Modeling skylight and aerial perspective". En *ACM SIGGRAPH 2003 Course Notes* (2003), ATI Research.
- Pribram, K. 1971. *Languages of the brain. Experimental paradoxes and principles in neuropsychology*. New York, Prentice-Hall.
- Pritchard, D.C. 1999. *Lighting*. Addison Wesley. 6ª edición. Primera edición, 1969.
- Purcell T. J.; Buck I.; Mark W. R.; Hanrahan, P. 2002. "Ray tracing on programmable graphics hardware". En *Proceedings of siggraph* (2002), vol 21, p 703.
- Purcell T. J.; Donner C.; Cammarano M.; Jensen H. W., Hanrahan P. 2003. "Photon mapping on programmable graphics hardware". En *Proceedings of Graphics Hardware* (2003), pp 41–50.
- Radiance. Ver la página oficial: <http://radsite.lbl.gov/radiance/HOME.html>.
- Ramamoorthi, R. 2009. "Precomputation-Based Rendering". En: *Foundations and Trends in Computer Graphics and Vision*, vol 3, n 4 (2007), pp 281-369.
- Rayleigh, L. 1871. "On the scattering of light by small particles". *Philosophical Magazine* 41 (1871), 447–454.
- Reichert, M. C. 1992. *A two-pass radiosity method driven by lights and viewer position*. Master's thesis, Cornell University.
- Reinhard, E.; Stark, M.; Shirley, P.; Ferwerda, J. 2002. "Photographic tone reproduction for digital images". *ACM Trans. Graph. (special issue SIGGRAPH 2002)* 21, 3, pp 267-276.
- Reinhard, E.; Ward, G.; Pattanaik, S.; Debevec, P. 2005. *High Dynamic Range Imaging - Acquisition, Display and Image-Based Lighting*, Elsevier.
- Ritschel, T.; Engelhardt, T.; Grosch, T.; Seidel H. P.; Kautz J.; Dachsbacher, C. 2009. "Micro-rendering for scalable, parallel final gathering. *ACM Transactions on Graphics* 28, 5 (2009), 132:1–132:8. (*Proc. Siggraph Asia*).
- Ritschel, T.; Dachsbacher, C.; Grosch, T.; Kautz, K. 2012. "The State of the Art in Interactive Global Illumination". *Computer Graphics Forum*, Volume 31 (2012), n 1, february, pp 160-188.
- Robbins, C.L. 1986. *Daylighting. Design & Analysis*, New York, Van Nostrand Reinhold.
- Ross, S. M. 2002. *Introduction to Probability Models* (8th ed.). San Diego: Academic Press.
- Schlick, C. 1991. "An Adaptive Sampling Technique for Multidimensional Integrator by Ray Tracing". En *Second Eurographics Workshop on Rendering* (Spain), pp 48-56
- Schumacker, R., Sproyll, R.F. & Sutherland, I.E. 1974. "A characterization of ten hidden-surface algorithms". *Computings Surveys*, 6 (1), pp 1-55.
- Segovia B.; lehl J. C.; Mitanchey R.; Péroche B. 2006. "Bidirectional instant radiosity". En *Proceedings of EGSR* (2006), pp 389–398.
- Segovia, B; lehl, J.C.; Péroche, B. 2007 "Metropolis Instant Radiosity". *Eurographics 2007*.



- Seetzen, H.; Heidrich, W.; Ward, G.; et al. 2004. "High dynamic range display systems". En *Proc. of ACM SIGGRAPH 2004*.
- Shanmugan, P.; Arikan, O. 2007. "Hardware accelerated ambient occlusion techniques on GPUs". En: *Proceedings SI3D '07*, pp 73-80.
- Shirley, P., B. Wade, P. Hubbard, D. Zareski, B. Walter, and D. P. Greenberg. 1995. "Global illumination via density estimation". *Eurographics Rendering Workshop 1995*, pp 219-31.
- Sillion, F.X., Puech, C. 1989. "A General Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods". *Computer Graphics (Proc. Siggraph)*, vol 23, n 3, july, pp 335-344.
- Sloan P.; Kautz J.; Snyder J. 2002. "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments". *ACM Transactions on Graphics* 21, 3 (2002), pp 527-536. (Proc. Siggraph).
- Stevens, S. S. 1946. "On the Theory of Scales of Measurement". *Science* 103 (2684), pp 677-680.
- Stevens, S.S. 1961. "To Honor Fechner and Repeal His Law". *Science*, vol 133, january, pp 80-86.
- Sutherland, I.E.; Sproull, R.F.; Schumacker, R.A. 1974. "A Characterization of Ten Hidden-Surface Algorithms". *ACM Computing Surveys*, 6(1), march 1974, pp 1-55.
- Tregenza, P. R. 1983. "The Monte Carlo method in lighting calculations". *Lighting Research and Technology* 15(4), pp 163-70.
- Torrance, K.E., Sparrow, E.M. 1967. "Theory for off-specular reflection from roughened surfaces". *Optical Society of America*, 57 (9), pp 1105-14.
- Veach, E.; Guibas, L. J. 1994. "Bidirectional Estimators for Light Transport". *Proceedings of Fifth Eurographics Workshop on Rendering*.
- Veach, E.; Guibas, L. J. 1995. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". *Proceedings of SIGGRAPH 95*.
- Veach, E. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. thesis, Stanford University.
- Veach, E., and L. J. Guibas. 1997. "Metropolis light transport". *Computer Graphics (SIGGRAPH '97 Proceedings)*, pp 65-76.
- Vos, J. 1984. "Disability glare - a state of the art report". *CIE Journal* 3, 2, 39-53.
- Wald, I.; Kollig, T.; Benthin, C; Keller, A.; Slusallek, P. 2002. "Interactive global Illumination using fast ray tracing". En: *Proceedings of EGWR (2002)*, pp 15-24.
- Wald, I.; Havran, V. 2006. "On building fast kd-trees for Ray Tracing and on doing that in  $O(N \log N)$ ". *Tech. Rep. UUSCI-2006-009*, SCI Institute, University of Utah.
- Wallace, J.R.; Cohen, M.F.; Greenberg, D.P. 1987. "A two-pass solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods". *SIGGRAPH 87*, pp 331-320.
- Walter, B.; Fernandez, S.; Arbre, A.; Bala, K.; Donikian, M.; Greenberg, D. P. 2005. "Lightcuts: a scalable approach to illumination". *ACM Transactions on Graphics* 24, 3 (2005), pp 1098-1107. (Proc. Siggraph).
- Wandell, B.A. 1995. *Foundations of Vision*. Sinauer Ass., Massachusetts.
- Wang, R.; Tran, J.; Luebke, D. 2004. "All-frequency relighting of non-diffuse objects using separable BRDF approximation". En *Proceedings of EGSR*.
- Wang, R.; Tran, J.; Luebke, D. 2006. "All-frequency relighting of glossy objects". *ACM Transactions on Graphics* 25, 2 (2006), pp 293-318.
- Wang, R.; Wang, R.; Zhou, K.; Pan, M.; Bao, H. 2009. "An efficient GPU-based approach for interactive global illumination". *ACM Transactions on Graphics* 28, 3 (2009), 91:1-91:8. (Proc. Siggraph).
- Ward, G. "High Dynamic Range Image Encodings" *Anywhere Software*.
- Ward, G.J.; Rubinstein, F.M.; Clear, R.D. 1988. "A Ray Tracing Solution for Diffuse Interreflection". *Computer Graphics*, vol 22, n 4, august.
- Ward, G.J. 1994. "The RADIANCE Lighting Simulation and Rendering System". *Computer Graphics Proceedings*, pp 459-472.
- Warnock, J. 1969. "A Hidden Surface Algorithm for Computer Generated Halfone Pictures". *University of Utah, Computer Science Dept. C.S. Tech. Report 4-15*.
- Watkins, G.S. 1970. "A Real Time Visible Surface Algorithm". *University of Utah Computer Science Dept., Tech. Rep. UTEC-CSC-70-101*, june 1970.

- Whitted, J.T. 1978. "A scan line algorithm for the computer display of curved surfaces". *Proc. 5th Conf. on Computer Graphics and Interactive Techniques*. Atlanta GA, p.2.
- Whitted, J.T. 1980. "An improved illumination model for shaded display". *Comm. ACM*, 23 (6), pp 342-9.
- Williams, L. "Casting Curved Shadows on Curved Surfaces". *Siggraph 78*, pp 270-274.
- Wyszecki, G., Stiles, W.S. 1982. *Color Science. Concepts and Methods, Quantitative Data and Formulae*. Wiley-Interscience publication.
- Zhou K.; Hou Q.; Wang R.; Guo B. 2008. "Real-time KD-tree construction on graphics hardware". *ACM Transactions on Graphics* 27, 5 (2008), 126:1–126:11. (*Proc. SIGGRAPH Asia*).
- Zhukov, S., Iones, A., Kronin, G. 1998. "An ambient light illumination model". En: Drettakis, G., Max, N. (eds.) *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pp 45–56. Springer, Berlin Heidelberg NewYork.





# Índice analítico

## A

Adams 87, 138. Véase también Zonas, sistema de  
Adaptive sampling 166, 228, 354, 367  
Airy 63. Véase también Ojo, círculos de confusión  
Aliasing, Antialiasing 146, 147, 160, 161, 167,  
172, 187, 200, 207, 210, 214, 215, 223,  
226–231, 232, 242, 304, 333, 334, 335, 341,  
351, 360, 364, 368, 370, 376, 381, 386, 408,  
415, 416, 426  
Ambient Occlusion 143, 187–189, 417, 422–426  
con la GPU 220–223  
Apple 103, 145, 146  
Árboles kd, árboles octales. Véase kd-trees,  
octrees  
Arikan 221, 226  
Arvo 159, 189  
Ashikhmin 137, 138, 155

## B

Bandas de Mach 85–86  
Basquin 31  
Battaille 142, 147  
Biased, Unbiased 143, 152, 165, 171, 184, 341,  
343, 349, 409  
Blinn 93, 155, 242  
Bohm 57  
Bokeh 235  
Bouguer 31  
Bouknight 93  
BRDF. Véase Reflectancia  
Brick Maps 191  
Brillo. Véase Luminosidad  
BSP (Binary Space Partition). Véase Subdivisión  
espacial  
Bunell 220, 221  
BVH (Bounding Volume Hierarchies) 171, 216,  
218

## C

Cámaras  
reales 113, 116  
virtuales 107, 108, 234, 350  
Campbell y Gubisch 64  
Candela. Véase Fotometría, intensidad luminosa  
Carpenter 146, 339  
Carr 216  
Chevreul 83  
Chiu 137, 138

Choudhury 137  
Christensen 190, 192, 224, 225  
CIBSE (Chartered Institution of Building Services  
Engineers) 98, 299  
CIE 24, 32, 33, 35, 44, 69, 84, 100, 134, 135, 313,  
314, 315, 316, 28  
Cohen 150, 172  
Contraste 44, 91  
CSF (Contrast Sensitivity Function) 71  
cuantificación 69, 69–71, 126  
factor de 80, 113, 121, 122, 124, 128  
umbral 229  
Cook 146, 155, 169, 189, 200, 339  
Crow 101, 102, 226, 307  
CSF (Contrast Sensitivity Function). Véase Contraste  
Curva de Reproducción de Tono 128

## D

Dachsbacher 212, 215  
Debevec 132, 134, 137, 143, 185, 186, 240, 242,  
245, 276, 347  
Densitometría 117, 123–126  
Dersch 135, 246  
DF (Daylight Factor). Véase Factor de luz diurna  
Direct3D 92, 199–202, 203, 204, 205, 251  
Dispersión (scattering) 36–38  
DMC (Deterministic Monte Carlo). Véase Ray  
tracing, Montecarlo  
Dorsey 137, 138, 277  
Drago 138, 277, 278  
Durand 137, 138, 277  
Dutre 160

## E

Ecuación de rendering. Véase Rendering Equation  
Entorno, mapas de. Véase Environment mapping  
Environment mapping 184, 242, 250, 319, 403  
Escalas de valores 81–84  
CIE (lightness function) 84  
Foss, Nickerson y Granville 85  
ley de Stevens 83  
Munsell, Sloan y Godlove 84  
Newhall, Nickerson y Judd 84  
Priest, Gibson y McNichols 84  
Richter, DIN 85  
EULUMDAT 98, 299  
EV (Exposure Value) 87, 109, 114, 115, 116, 117,  
118, 123, 124, 126, 134, 135, 235, 237, 238,  
239, 241, 254, 259, 282, 284, 308, 309, 312,





314, 321  
Exposición, control de 87, 109, 112, 235, 241,  
282–285, 297, 316, 317, 321, 322, 328, 359,  
404. Véase también EV (exposure value)  
EXR. Véase HDR, formatos

## F

Factor de luz diurna 29, 431  
Fattal 137, 138  
Fechner 67, 68, 83  
Filtros, convolución 232  
Final Gather 179–181, 328, 341, 356, 357, 427.  
Véase también Irradiancia, irradiancia y final  
gather  
con IBL 407, 409, 411, 414, 416, 419, 420  
con mental ray 376–382, 398  
con radiossidad (regathering) 150  
e Irradiance con la GPU 222–226  
Flujo luminoso. Véase Fotometría, flujo  
Fotometría. Véase también Tipos de luces, luces  
fotométricas  
apostilib 25  
eficacia luminosa 22, 24  
exitancia 22  
flujo, lúmenes 22, 23, 24, 26, 30, 90, 96, 295,  
296, 299  
iluminancia, luxes 22, 23, 24, 26, 28, 30, 32,  
35, 77, 89, 91, 96, 116, 122, 282, 296, 311,  
314, 427, 429, 430, 433  
intensidad luminosa, candelas 22, 23, 24, 25,  
26, 90, 96, 296  
luminancia 22, 27, 357, 427  
mililamberts 25  
unidades 22, 90, 132, 149, 295, 352  
Fotones  
cámaras digitales 113, 114, 115, 127  
con final gather 180, 181, 341  
con importons 190, 400  
con mental ray 383–388  
con V-Ray 387–391  
emisión virtual 164  
proyección (photon mapping) 144, 176–178,  
208, 320, 325, 334, 382  
proyección (photon mapping) con la GPU 218, 219  
reutilización del mapa de fotones 397–400  
y cáusticas 178, 179, 392–398  
Fourier 72, 73, 74, 213, 214, 226, 232  
Frame buffer 108, 129, 198, 217  
Función de respuesta 160

## G

Gabor 58  
Gamma 125, 126–129, 134, 137, 138, 276, 278,  
279, 279–281, 283, 404, 407  
Glare. Véase Resplandor

Glow. Véase Resplandor  
Goethe 21, 48  
Goldstein 145, 146  
Goral 142, 147  
Gouraud 93, 205  
GPU (Graphic Processing Unit) 197–199,  
206–209, 232, 336  
AMD 197, 199, 201, 336, 347  
ATI. Véase AMD  
CUDA 199, 201, 225, 336, 341, 345, 348, 358  
iluminación global 206–227  
Nvidia 189, 199, 242, 250, 336, 337, 340, 341,  
347, 348, 358  
Greenberg 97, 142, 147, 150, 172, 189  
Greene 242  
Grosseteste 21  
Guibas 172, 182

## H

Hachisuka 223  
Hanrahan 159, 216, 339  
Hasan 226  
HDR (High Dynamic Range). Véase también IBL  
(Image Based Lighting)  
edición 270–273  
formatos, hdr, exr 132–138, 135, 136, 240,  
260, 262, 266, 268, 402, 405, 407, 409, 410,  
411, 413, 421  
métodos de generación 240–242  
métodos de obtención 236–239  
panorámicos, generación 260–263  
panorámicos, obtención por bola especular  
255–258  
panorámicos, obtención por cámaras especia-  
les 259  
panorámicos, obtención por series 251–255  
panorámicos, tipos 241, 245  
panorámicos, transformación 247–249  
panorámicos, virtuales 262–270  
reproyección de valores LDR 276–279  
HDR Shop 240, 245, 246, 248, 257, 262, 266,  
270, 408  
Heckbert 142, 159, 174  
Heidmann 101, 307  
Hemisferio. Véase también Skydome  
celeste 32, 35, 313  
coordenadas 153  
en cálculos de iluminación global 150, 152,  
157, 158, 164, 168, 175, 189, 223, 225, 371,  
377, 423  
envolvente 186, 243, 249, 403, 413, 414, 421, 422  
Hoberock 222  
Hoffman 242  
Hologramas 58  
Hopkinson 30, 34, 36

Hugin 243, 246, 248, 260  
Hurter y Driffeld, curvas H-D 124. Véase también Densitometría

## I

IBL (Image Based Lighting) 99, 133, 143, 184–187, 236, 242, 245, 258, 336, 408, 421  
configuraciones 401–410  
ejemplos 411–420  
IBR, Image Based Rendering 185  
IES (Illuminating Engineering Society) 97, 98, 297–301, 311, 313, 324  
IESNA (Illuminating Engineering Society of North America) 98  
Iluminación. Véase también Fotometría, Luces, Rendering equation  
calidad lumínica 44, 89, 91  
directa, indirecta 167, 168, 287, 291, 333  
global 104, 106, 133, 141, 142–145, 333  
local 142, 202, 204, 333, 335  
medición 425–433  
Image Based Modelling 185  
Importance function. Véase Función de respuesta  
Importance Sampling 152, 161, 165, 166, 168, 169, 173, 224, 225, 340, 341, 342, 349, 377, 421  
Instant Radiosity 209  
Intensidad luminosa. Véase Fotometría, intensidad luminosa  
Irradiancia 22, 26, 27, 29, 153, 155  
con V-Ray 368–375  
irradiance caching 142, 143, 173, 176, 181  
irradiance particles 191  
irradiancia y final gather 222, 371  
irradiancia y fotones 382, 386, 390  
mapa de irradiancia 398, 400

## J

Jensen 142, 172, 176, 183, 190, 218

## K

Kajiya 142, 147, 150, 151, 152, 156, 175, 347  
Kautz 213, 215  
Kay 145, 146  
Kd-tree 177, 217, 218, 219, 223  
Keller 209, 210, 212  
Kettler 32  
Kimball 31  
Kittler 32, 33, 34, 35  
Krivánek y Gautron 226

## L

Lafortune 155, 172, 190  
Landis 188  
LDR (Low Dynamic Range). Véase HDR (High

Dynamic Range)  
Leonardo da Vinci 38  
Lightcuts 209  
Light Probe 214, 245, 248, 249, 266, 402  
LSF (Line Spread Function) 74  
LTLI (Danish Illuminating Laboratory) 98, 299  
Lucas 135, 188, 200, 243, 339  
Luces  
ambiente 93, 289, 313  
atenuación 94, 291, 292  
celeste (skylight) 31, 99, 100, 236, 308, 312, 313  
de área, de volumen. Véase extensas  
diagrama goniométrico, diagrama polar 41, 42, 98  
direccionales 92, 205, 288, 290  
diurna 28, 31, 36, 78, 98, 100, 116, 282, 283, 308, 308–321, 321, 322, 429, 430  
extensas 47, 96, 101, 293, 294, 306, 307, 321, 334, 335  
focales, spot 92, 205, 288, 290, 291  
fotométricas 96, 97, 294–301  
negativas 105  
nocturna 40  
puntuales, omnidireccionales 92, 205, 288, 289  
simples, standard 28, 92–96, 288, 296  
solar 29, 30, 32, 37, 60, 68, 76, 98–100, 311  
unidades de intensidad 90, 106, 289, 311, 313, 321, 322, 351. Véase también Fotometría, unidades  
Luminance HDR 241  
Luminancia. Véase Fotometría, luminancia  
Luminarias 42, 43, 100, 323, 325  
Luminosidad 28, 84, 127

## M

Mach 85. Véase también Bandas de Mach  
Maletz 225  
Markov 162, 164, 182  
Material de superposición (override material) 233  
McGuire y Luebke 219  
Measurement equation 161  
Metropolis-Hastings 182  
Micro-buffers 224, 225  
Mie 38  
Miller 242  
Minnaert 39, 77  
MIS (Multiple Importance Sampling) 341, 344, 347, 348  
MLT (Metropolis Light Transport) 173, 182–184, 212, 346  
Montecarlo. Véase Ray tracing, Montecarlo  
Moon y Spencer 32, 33, 80  
mr Physical Sky 284, 308, 312, 313, 320, 322



mr sky 284, 312, 313, 316, 320, 414  
mr Sun 310, 311, 313, 359, 362, 387, 411, 427  
MTF (Modulation Transfer Function) 71, 73  
Multiplicador. Véase Luces, unidades de intensidad  
Munsell 84

## N

Nagel 145, 146  
Newell 242  
Nishita 101

## O

Oclusión ambiental. Véase Ambient Occlusion  
Octrees 171, 192, 217, 218  
Ojo. Sistema visual  
    adaptación 66  
    agudeza visual 59–63, 64  
    círculos de confusión, de Airy 62  
    córtex visual 55–58  
    discriminación de luminancias 67–71  
    dispersión en el ojo, difracción 64  
    estructura 49–52  
    fóvea, tamaño, ángulos 50  
    mecanismos neuronales 52  
OpenEXR. Véase HDR (High Dynamic Range)  
OpenGL 92, 197, 199–203, 204, 205  
OTF (Optical Transfer Function) 72, 74

## P

Panoramas. Véase HDR panorámicos, tipos  
Path tracing 156  
    bidireccional 172  
    con iray 358–363  
    con V-Ray 363–368  
Pattanaik 137, 138, 190  
PDF (Probability Distribution Function) 163, 164, 173, 183  
Perez 32, 35, 36, 100, 313–317, 428  
Perspectiva aérea 38, 317  
Pharr 183  
Phong 93, 155, 211  
Photon mapping. Véase Fotones, proyección  
Picturenaut 240, 260, 262, 266, 270, 277, 278, 402, 418  
Pixar 136, 187, 189, 200, 201, 339, 342, 345  
Pleijel 31, 35  
Point clouds, Point based GI 187, 191, 192, 225  
Pokrowski 32, 34  
Porter 146  
Poynton 128  
Pribram 57  
Progressive rendering 230, 231, 338, 364, 367, 401, 408, 416, 417  
PSF (Point Spread Function) 74  
PTF (Phase Transfer Function) 74

PTGUI 246, 248, 260  
Purcell 216, 218, 219, 223

## Q

QMC (Quasi Monte Carlo). Véase Ray tracing, Montecarlo

## R

Radiancia 22, 26, 27, 153, 154, 155, 225. Véase *también* importance function, rendering equation, path tracing  
    celeste 31  
    función de transferencia de 214  
    Radiance (software) 134, 142, 175  
    radiancia incidente, cálculo 160, 168  
    radiancia y fotones, estimación 177, 178  
Radiometría. Véase *también* Radiancia, Irradiancia  
    energía radiante 21, 22  
    exitancia radiante 22  
    intensidad radiante 22  
    potencia radiante 22  
Radiosidad 147–150  
    con 3ds Max 351–356, 427  
    con otros programas 357  
    con ray tracing 172  
    jerárquica 149  
Rangos dinámicos. Véase *también* HDR (High Dynamic Range)  
    decibelio 79, 80  
    de dispositivos de entrada (cámaras, escáners) 113–119  
    de dispositivos de salida (monitores, impresoras) 120–126  
    escotópico 77  
    fotópico 78  
    mesópico 78  
    umbrales inferiores 74  
    umbrales superiores 77  
Rasterización 192, 201, 204, 215, 216, 223  
    versus ray tracing 206–209  
Rayleigh 32, 37, 38, 63  
Ray Tracing. Véase *también* Path Tracing, Rendering Equation  
    clásico, Whitted-style, recursive 142, 145, 146, 169  
    con la GPU 215–219  
    con radiosidad 172  
    distribuido 147, 169, 176  
    forward tracing, backward tracing 157–160  
    Montecarlo, estocástico, generalizado 161, 169–171, 368–371  
    sombras 103  
    versus Rasterization 206–209  
Reflectancia 27, 28, 32, 44, 48, 53, 70, 84, 87, 107, 112, 117, 122, 124, 151, 168, 185, 233,

257, 258, 356, 357, 364, 382, 387, 392, 421, 427, 428, 429, 433  
función de reflectancia, BRDF 154–156, 159, 169, 173  
Reinhard 137, 138, 276  
Rendering Equation 142, 150–153, 156, 158, 161, 213, 347, 391  
RenderMan 187, 191, 200, 201, 336, 337, 338, 339, 341, 342, 343, 344, 346, 347  
Resolución 59–64, 71, 109–112, 144, 197, 198, 207, 227, 228, 230, 231, 233, 236, 254, 259, 263, 334, 335, 402, 409, 417–421  
Resplandor 26, 43, 80, 328, 329  
Reyes (software) 192, 339, 340, 343, 344  
Ritschel 215  
Rushmeier 137

## S

Sampling 165, 166, 228. *Véase también* adaptive sampling, importance sampling, stratified sampling, unified sampling  
Scattering. *Véase* Dispersión, Rayleigh, Mie  
Sensitometría. *Véase* Densitometría  
Shader 159, 169, 186, 200, 206, 323, 338, 339, 341, 343, 359, 377, 392, 393, 400, 401, 403, 408, 411, 419, 422, 426  
de Terragen 263–265  
fragment shader. *Véase* Shader, pixel shader  
pixel shader 93, 198, 201, 205, 207, 212, 216, 217, 219, 220, 221  
vertex shader 93, 198, 201, 216  
Shirley 143, 155  
SH (Spherical Harmonics) 213, 214, 215, 224, 226  
Skybox 243, 249, 265  
Skydome 243, 249  
Skylight. *Véase* Luces, celeste; *Véase también* IBL (Image Based Lighting)  
Sky portals 320–322, 429  
Sloan 213, 214, 215  
Snyder 213  
Sombras  
calidad y función visual 44–48  
extensas, de área 104, 306  
mapa de sombra (shadow map) 102, 302–305  
suavizadas 104  
tipo de cálculo, parámetros generales 301, 302, 334  
transparentes 105, 305  
trazado de rayos (ray traced shadows) 103, 145, 302, 304  
umbra y penumbra 95, 101, 105, 147, 149, 291, 293, 294, 307, 311, 324  
volumen de sombra (shadow volume) 101, 307  
Stamminger 212  
Stencil buffer 204, 217, 219, 307

Stevens 83. *Véase también* Escalas de valores  
Stitching 186, 237, 243, 246, 247, 251, 259, 260.  
*Véase también* HDR, panorámicos  
Stratified Sampling 165, 168  
Subdivisión espacial 170, 171, 216  
Surfels 192, 220, 221, 224

## T

Temperatura de color 44, 91, 97, 283, 297, 311, 313, 323  
Terragen 262–265  
Tone mapping 90, 123, 134, 137, 240, 276, 282  
Torrance 142, 147, 155  
TRC (Tone Reproduction Curve). *Véase* Curva de reproducción de tono  
Troland 78  
Trotter 29  
Tumblin 137  
Turk 137  
Turner 31

## U

Ulam 161  
Unified sampling 230, 408

## V

Valenciennes 38  
Veach 165, 172, 173, 182, 183  
Verbeck y Greenberg 97  
Vernier 64. *Véase también* Ojo, agudeza visual  
Volumen de luz 324, 328  
Von Neumann 161, 163  
Voxels. *Véase* Subdivisión espacial  
VPLs (Virtual Point Lights) 209

## W

Waldram 29, 31  
Wallace 172  
Walter 226  
Wang 218, 225  
Ward 132, 134, 135, 137, 142, 143, 155, 173, 174, 175, 185, 189, 211, 226, 276  
Wavelet transform 233  
Weber 67, 68, 70, 85, 211  
Weston 87. *Véase también* Zonas, sistema de  
Whitted 142, 146, 169, 175, 216, 217  
Wyszecki y Stiles 50, 68, 78, 84

## Z

Z-buffer 102, 204, 221  
Zhou 219  
Zhukov 188  
Zonas, sistema de 86–88. *Véase también* Escalas de valores







